



TUGAS AKHIR - TE 141599

MULTI OBJEKTIF OPTIMAL POWER FLOW UNTUK MINIMISASI BIAYA OPERASI DAN EMISI KARBON MENGGUNAKAN ALGORITMA CUCKOO

Herbanindya Febri Trusatmaji
NRP 2210100015

Dosen Pembimbing
Prof. Ir. H. Ontoseno Penangsang, M.Sc, Ph.D.
Yun Tonce Kusuma Priyanto, ST, MT

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 141599

**MULTI OBJECTIVE OPTIMAL POWER FLOW TO
MINIMIZE GENERATION OPERATION COST AND
CARBON EMISSION USING CUCKOO ALGORITHM**

Herbanindya Febri T
NRP 2210100015

Advisor
Prof. Ir. H. Ontoseno Penangsang, M.Sc, Ph.D
Yun Tonce Kusuma Priyanto, ST, MT

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty Of Industrial Technology
Sepuluh Nopember Institute Of Technology
Surabaya 2016

**Multi Objektif Optimal Power Flow untuk Minimisasi Biaya
Operasi dan Emisi pada Pembangkit Menggunakan Algoritma
Cuckoo**

TUGAS AKHIR

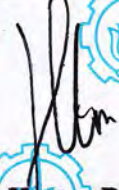
**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada Bidang Studi Teknik Sistem Tenaga
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,



Dosen Pembimbing II,



Prof. Ir. H. Ontoseno Penangesang, M.Sc, Ph.D
NIP. 1949 07 15 1974 12 1001

Yun Tonce Kusuma Privanto, ST, MT.
NIP. 1984 06 16 2012 12 1001



MULTI OBJEKTIF OPTIMAL POWER FLOW UNTUK MINIMISASI BIAYA OPERASI DAN EMISI PADA PEMBANGKIT MENGGUNAKAN ALGORITMA CUCKOO

Nama Mahasiswa : Herbanindya FT
NRP : 2210100015
Dosen Pembimbing : Prof. Ir. H. Ontoseno Penangsang, M.Sc,Ph.D
: Yun Tonce Kusuma Priyanto, ST, MT

ABSTRAK

Dalam sistem tenaga listrik, pembangkitan energi listrik selalu berkaitan dengan bahan bakar yang digunakan. Sebagian besar pembangkit tenaga uap menggunakan bahan bakar batubara . Dengan kata lain semakin energi yang dihasilkan, semakin tinggi pula bahan bakar yang digunakan yaitu batubara. Hal ini menyebabkan tingginya isu lingkungan yang diakibatkan oleh sisa pembakaran batubara.

Jumlah polutan hasil pembakaran yang dihasilkan oleh pembangkit bertambah seiring dengan berjalannya waktu dan masih belum ditemukan teknologi yang mampu menyelesaikan permasalahan secara menyeluruh. Oleh karena itu, untuk menyelesaikan ketidakseimbangan antara pertumbuhan beban , penambahan jumlah pembangkit dan polutan yang dihasilkan dari pusat pembangkit listrik, diperlukan sebuah metode untuk memperoleh energi listrik dengan kadar emisi seoptimal mungkin

Untuk memaksimalkan daya pembangkitan dengan biaya yang rendah dan menekan jumlah polutan yang dihasilkan dari pusat pembangkit listrik. Dilakukan simulasi *Multi Objective Optimal Power Flow* menggunakan algoritma Cuckoo pada *software* MATLAB.

Kata kunci : *Multiobjective Optimal Power Flow* ,Emisi Karbon,
Biaya Pembangkitan

Halaman ini sengaja dikosongkan

**MULTIOBJECTIVE OPTIMAL POWER FLOW TO MINIMIZE
GENERATION OPERATION COST AND CARBON EMISSION
USING CUCKOO ALGORITHM**

Name : Herbanindya FT
NRP : 2210100015
Supervisor : Prof. Ir. H. Ontoseno Penangsang, M.Sc,Ph.D
: Yun Tonce Kusuma Priyanto, ST, MT

ABSTRACT

In electric power systems , electrical energy generation is always associated with fuel . Most steam power plants using coal as their fuel . In other words, the more energy is generated, then higher amount the fuel used - coal . This leads to higher environmental issues that caused by the combustion of the coal .

The amount of pollutant as the residu of the generator combustion increase significantly following by the time and there is no invention of new technology to solve this problems. Therefore, to solve the unbalancing condition between the load grow, the addition of new generator , and the residu of generator combustion, urgently needed a new method to get electrical power with emission rate as optimize as possible.

Based on those problems, to maximize electrical power with the lowest generating cost and emission from generator combustion, make a new innovation which named as “multiobjective optimal power flow to minimize generation operation cost and carbon emission using cuckoo algorithm” with software MATLAB

Key Words : Multiobjective Optimal Power Flow ,Carbon Emission, Generating Cost

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji kehadiran Tuhan Yang Maha Esa. atas karunia, kesehatan, dan kesempatan awal hingga akhir pengerjaan tugas akhir ini. Puji syukur senantiasa saya hadirkan kepada Nabi Muhammad saw. Secara khusus, penulis ingin mengucapkan terima kasih serta apresiasi yang sangat mendalam kepada pihak-pihak yang telah membantu mendukung penyelesaian tugas akhir ini, diantaranya :

1. Kedua orang tua, Wilarso Hermanto dan Sri Wibawanti serta keluarga, atas do'a, nasihat, dukungan serta motivasi yang telah diberikan.
2. Bapak Ontoseno Penansang, selaku dosen pembimbing pertama, atas kesediaannya memberikan masukan dan waktu untuk menyelesaikan tugas akhir ini.
3. Bapak Yun Tonce selaku dosen pembimbing kedua, atas bimbingan dan penjelasan serta motivasi yang diberikan.
4. Bapak Heri Suryo, selaku dosen wali yang telah memberikan saran-saran pada perkuliahan dan beberapa perencanaan akademik.
5. Seluruh dosen bidang studi sistem tenaga, atas bimbingan serta ilmu-ilmu perkuliahan yang diberikan.
6. Seluruh rekan-rekan angkatan 2010 yang selalu menyemangati untuk terus kuliah.

Besar harapan penulis agar tugas akhir ini dapat memberikan manfaat dan masukan bagi banyak pihak. Oleh karena itu penulis mengharapkan kritik, koreksi, dan saran dari pembaca yang bersifat membangun untuk pengembangan kearah yang lebih baik.

Surabaya , Januari 2016

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

Halaman

TITTLE	
FINAL PROJECT STATEMENT SHEET	
APPROVAL SHEET	
ABSTRAK	i
ABSTRACT	iii
INTRODUCTION	v
TABLE OF CONTENTS	vii
LIST OF FIGURE	xi
LIST OF TABLE	xiii
 CHAPTER 1 PREFACE	 1
1.1 Background.....	1
1.2 Problems	2
1.3 Problems Boundary.....	3
1.4 Purpose.....	3
1.5 Systemathics.....	3
 CHAPTER II LITERATURE	 5
2.1 Power System.....	5
2.1.1 Electric Power Generator.....	5
2.1.2 Transmision and Distribution.....	8
2.1.3 Load.....	9
2.2 Power Flow Analysis.....	9
2.3 Mathematics Equation of Power Flow.....	11
2.3.1 Inductive Load.....	14
2.3.2 Capacitive Load.....	15
2.3.3 Power Flow Analysis	
Newton Rhapson Method.....	16
2.4 Optimal Power Flow.....	18
2.4.1 Active Power Dispatch	19
2.4.1.1 Operation Cost Function.....	19
2.4.1.2 Carbon Emission Function.....	20
2.5 Optimal Power Flow Application.....	21
2.6. Particle Swarm Optimization.....	21
2.7 Cuckoo Search Algorithm.....	24
2.7.1 Host Search Equation	26

CHAPTER III METODOLOGY	27
3.1 Research Plan	27
3.2 Search for Books and Literature	28
3.3 Data Processing	28
3.4 Objective Function	28
3.4.1 Carbon Emission Function	28
3.4.2 Generation Cost Function	29
3.5 <i>Constraint</i>	29
3.5.1 Power Generator Constraint	29
3.5.2 Power Balance	31
3.6 Divide Objective Fuction With Weighted Sum Method	31
3.7 Cuckoo Search Algorithm Simulation	32
3.7.1 Initialization Cuckoo	34
3.7.2 Moving Cuckoo (Levy Flight)	34
3.8 Particle Swarm Optimization (PSO) Simulation	34
3.8.1 Particle Swarm Optimization	36
 CHAPTER IV SIMULATION AND ANALYSIS	 39
4.1 IEEE 30 Bus Data	39
4.2 <i>Optimal Power Flow Economic Dispatch Simulation</i>	42
4.2.1 OPF Economic Dispatch Using Cuckoo Search Algorithm Simulation	42
4.2.2 OPF Economic Dispatch Using Particle Swarm Optimization (PSO) Simulation	44
4.2.3 Comparation Of OPF Economic Dispatch Using Cuckoo and PSO	45
4.3 <i>Optimal Power FlowEmission Minimization Simulation</i>	46
4.3.1 OPF Emission Minimization Using Cuckoo Search Algorithm Simulation	46
4.3.2 OPF Emission Minimization Using Particle Swarm Optimization (PSO) Simulation	48
4.3.3 Comparation Of OPF Emission Minimization Using Cuckoo Search Algorithm and Particle SwarmOptimization Method Simulation	50
4.4 Optimal Power FlowEconomic Emission Dispatch Sumulation	51
4.4.1 OPF Economic Emission Dispatchusing Cuckoo Search Algorithm Simulation	51

4.4.2 OPF Economic Emission Dispatch Using Particle Swarm Optimization (PSO) Simulation	52
4.4.3 Comparison of OPF Economic Emission Dispatch using Cuckoo Search Algorithm and Particle Swarm Optimization.....	53
CHAPTER V CONCLUSION AND RECOMMENDATION.....	55
5.1 Conclusion	55
5.2 Recommendation	55
REFERENCES.....	57
ATTACHMENT	59
BIOGRAPHY	72

DAFTAR GAMBAR

	Halaman
Gambar 3.1 Flowchart tahap – tahap penelitian <i>Optimal Power Flow</i>	27
Gambar 3.2 Flowchart penyelesaian Optimal Power Flow menggunakan Cuckoo	33
Gambar 3.3 Flowchart penyelesaian Optimal Power Flow menggunakan <i>Particle Swarm</i> <i>Optimization</i>	35
Gambar 4.1 Grafik Konvergensi OPF <i>Economic Dispatch</i> menggunakan metode Cuckoo ..	43
Gambar 4.2 Grafik Konvergensi OPF <i>Economic Dispatch</i> menggunakan Metode PSO	44
Gambar 4.3 Perbandingan Nilai Konvergensi Cuckoo dan PSO pada simulasi OPF <i>Economic Dispatch</i>	46
Gambar 4.4 Grafik Konvergensi OPF <i>Emission</i> <i>Minimization</i> menggunakan metode Cuckoo	47
Gambar 4.5 Grafik Konvergensi OPF <i>Emission Minimization</i> menggunakan metode PSO	49
Gambar 4.6 Perbandingan Nilai Konvergensi Cuckoo dan PSO pada simulasi OPF <i>Emission Minimization</i>	50
Gambar 4.7 Grafik konvergensi simulasi <i>economic emission</i> <i>dispatch</i> menggunakan metode Cuckoo	51
Gambar 4.8 Grafik konvergensi <i>economic emmission</i> <i>dispatch</i> menggunakan PSO	53
Gambar 4.9 Perbandingan grafik konvergensi <i>economic</i> <i>emmission dispatch</i> menggunakan metode Cuckoo dan PSO	54

Halaman ini sengaja dikosongkan

DAFTAR TABEL

	Halaman
Tabel 4.1 Data IEEE 30 Bus	39
Tabel 4.2 Data Saluran IEEE 30 Bus	40
Tabel 4.3 Data Generator IEEE 30 Bus	41
Tabel 4.4 Data Fungsi Biaya Pembangkitan Sistem IEEE 30 Bus	42
Tabel 4.5 Data Spesifikasi CPU	42
Tabel 4.6 Parameter Cuckoo	42
Tabel 4.7 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF Economic Dispatch menggunakan metode cuckoo	43
Tabel 4.8 Parameter PSO	44
Tabel 4.9 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF Economic Dispatch menggunakan metode cuckoo	45
Tabel 4.10 Perbandingan Hasil Simulasi OPF Economic Dispatch Menggunakan Metode Cuckoo dan PSO	45
Tabel 4.11 Parameter Cuckoo	46
Tabel 4.12 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF Emission Minimization menggunakan metode cuckoo	48
Tabel 4.13 Parameter PSO	48
Tabel 4.14 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF Emission Minimization menggunakan metode PSO	49
Tabel 4.15 Perbandingan Hasil Simulasi OPF Emission Minimization Menggunakan Metode Cuckoo dan PSO ..	50
Tabel 4.16 Parameter Cuckoo	51
Tabel 4.17 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF menggunakan metode cuckoo	52
Tabel 4.18 Parameter PSO	52
Tabel 4.19 Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF menggunakan metode cuckoo	53
Tabel 4.20 Perbandingan Hasil Simulasi OPF Economic Emission Dispatch Menggunakan metode Cuckoo Search dan PSO	54

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1. 1. Latar Belakang

Pada bidang sistem tenaga listrik modern, proses optimasi merupakan bagian yang penting. Sesuatu yang berkaitan dengan proses optimasi seperti rugi-rugi daya, besar tegangan bus, biaya pembangkitan dan daya yang diserap oleh beban merupakan parameter utama untuk menentukan besar daya yang akan dibangkitkan oleh pembangkit. Pada bidang sistem tenaga listrik semua komponen parameter dapat ditemukan melalui analisis aliran daya. Analisis aliran daya pada sistem tenaga, bertujuan untuk melihat bagaimana aliran daya dalam sistem. variabel yang terlibat dalam perhitungan adalah daya. Dengan menggunakan daya sebagai pengubah dalam perhitungan, maka persamaan yang kita hadapi menjadi bukan persamaan linier.

Ada beberapa parameter utama dalam proses optimasi yaitu efisiensi dan ekonomis. Yang dimaksud dalam hal ini adalah besar energi yang dibangkitkan sama atau mendekati besarnya permintaan energi listrik. Dengan cara menekan biaya pembangkitan dan rugi – rugi pada jaringan listrik. Dari waktu ke waktu permintaan jumlah energi listrik semakin meningkat, seiring dengan bertumbuhnya ekonomi dan bertambahnya jumlah penduduk. Hal tersebut tidak diiringi dengan tersedianya energi listrik yang cukup. Ini menyebabkan ketidakseimbangan antara permintaan dan produksi energi listrik. Oleh karena itu dibutuhkan analisi aliran daya agar didapatkan biaya pembangkitan yang minimal dan penyaluran energi listrik yang efisien serta tetap mempertimbangkan faktor emisi

Jarak antara pembangkit dan konsumen yang jauh menyebabkan rugi – rugi yang besar di saluran. Dengan adanya analisi aliran daya dapat diketahui besar rugi – rugi di setiap saluran. Serta efisiensi pada biaya pembangkitan harus dilakukan dengan metode *economic dispatch*. Metode ini dilakukan dengan mencari biaya pembangkitan setiap generator yang seminimal mungkin. Selain itu faktor emisi juga harus diperhatikan karena biaya pembangkitan yang murah identik dengan emisi yang tinggi. Dengan *software* MATLAB, dapat dilakukan simulasi untuk mendapatkan tingkat emisi yang baik dan biaya pembangkitan yang ekonomis dengan menggunakan *Cuckoo Search Algorithm*.

Algoritma cuckoo merupakan algoritma yang mengadaptasi dari cara berkembang biak burung Cuckoo. Dimana setiap telur yang

dihasilkan burung cuckoo dititipkan pada sarang burung lain untuk diletakkan. Sehingga burung cuckoo tidak perlu menetas telurnya dan membesarkan anaknya. Dengan cara ini probabilitas telur burung cuckoo untuk menetas menjadi besar. Cara unik inilah yang diadaptasikan menjadi algoritma yang cepat dan memiliki akurasi perhitungan yang tepat.

Algoritma burung Cuckoo atau *Cuckoo search Algorithm* merupakan algoritma metaheuristik yang relatif baru dibandingkan dengan metode optimisasi lainnya. Metode ini memiliki keunggulan dimana dapat mencapai hasil terbaik dengan kecepatan diatas algoritma lain. Dengan keunggulan tersebut, akan dibandingkan dengan algoritma PSO yang sudah lebih dulu dikenal karena kecepatannya dalam mencapai hasil terbaik.

Sebagai algoritma pembanding untuk menyelesaikan permasalahan multi objective optimal power flow, *Particle swarm optimization* (PSO) adalah teknik optimisasi stokastik berbasis populasi yang terkenal memiliki kecepatan dalam mencapai hasil yang terbaik. Dimana kehidupan di alam liar diadaptasikan ke dalam sebuah algoritma. Salah satu contohnya adalah kehidupan sekelompok ikan di alam liar yang bergerak bergerombol dan secara bersama – sama yang disebut dengan *swarm intelligence*. Algoritma ini yang telah dikembangkan oleh Dr. Eberhart dan Dr. Kennedy pada tahun 1995.

Dalam penelitian ini akan menunjukkan perbandingan hasil penyelesaian *Multi Objective Optimal Power Flow* dengan menggunakan metode *Cuckoo Search Algorithm* dengan metode *Particle Swarm Optimization* (PSO)

1. 2. Rumusan Permasalahan

Permasalahan yang akan dibahas dalam Tugas Akhir ini adalah :

- Bagaimana mengoptimalkan pengaturan daya pembangkit untuk meminimalisasi biaya dan emisi pada operasi pembangkitan?
- Bagaimana perbandingan fitness nilai terbaik dari metode *Cuckoo search Algorithm* dengan *Particle Swarm Optimization*?

1. 3. Batasan Permasalahan

Dalam penyelesaian masalah Tugas Akhir ini terdapat beberapa batasan – batasan dan asumsi sebagai berikut :

- Simulasi dilakukan dengan software MATLAB dengan plant IEEE 30-Bus. Algoritma yang digunakan *Cuckoo Search Algorithm* dan *Particle Swarm Optimization*.
- Fungsi objektif yang diatur secara optimal meliputi :
 - Fungsi emisi pembangkitan (*Emission*)
 - Fungsi biaya pembangkitan (*Economic Dispatch*)
- Penyelesaian Multi Objective Optimal Power Flow menggunakan dua program berbeda, yaitu :
 - *Cuckoo Search Algorithm*
 - *Particle Swarm Optimization* (PSO)

1. 4. Tujuan Permasalahan

Tujuan yang ingin dicapai dalam Tugas Akhir ini adalah :

- Untuk meminimalisasi emisi dan biaya operasi pada pembangkitan.
- Untuk mengetahui parameter nilai terbaik dari metode *Cuckoo Search Algorithm* dan *Particle Swarm Optimization*.

1. 5. Sistematika Penulisan

Penulisan tugas akhir ini akan dibagi menjadi lima bab dengan beberapa subbab. Berikut sistematika penulisannya :

Bab 1 : Pendahuluan

Dalam bab ini dibahas mengenai latar belakang, batasan masalah, tujuan, sistematika penulisan, dan metodologi.

Bab 2 : Tinjauan Pustaka

Dalam bab ini dibahas teori penunjang mengenai analisis aliran daya, *Cuckoo Search Algorithm* dan *Particle Swarm Optimization*.

Bab 3 : Metodologi Penelitian

Dalam bab ini diuraikan metode pengaplikasian konsep *Gravitational Cuckoo Search Algorithm* dan *Particle Swarm Optimization* pada OPF.

Bab 4 : Simulasi dan Analisis

Berisi tentang proses simulasi dan analisis hasil simulasi yang telah dilakukan.

Bab 5 : Kesimpulan dan Saran

Merupakan bab terakhir yang berisi tentang kesimpulan yang diperoleh dari hasil analisis simulasi yang dilakukan pada tugas akhir dan saran untuk pengembangan selanjutnya.

BAB 2

SISTEM TENAGA LISTRIK DAN STUDI ALIRAN DAYA

2. 1. Sistem Tenaga Listrik

Sistem bidang ketenagalistrikan, sistem tenaga listrik dapat dibagi menjadi 3 bagian penting yaitu:

- ❖ Pembangkitan Tenaga Listrik
- ❖ Transmisi jaringan listrik
- ❖ Distribusi jaringan listrik
- ❖ Beban

2. 1. 1. Pembangkit Tenaga Listrik

Pembangkit tenaga listrik adalah permulaan dimana tenaga listrik dihasilkan dari generator. Generator merupakan mesin listrik yang mengubah energi kinetik menjadi energi listrik. Sumber untuk mendapatkan energi kinetik terbagi menjadi dua kelompok yaitu energi Fosil dan energi terbarukan. Salah satu contoh energi terbarukan adalah seperti energi air dan matahari. Sementara untuk energi batubara, gas alam, bahan bakar minyak merupakan energi yang berasal dari fosil.

Berikut ini contoh pembangkit dengan sumber energi terbarukan:

1. Pembangkit Listrik Tenaga Air (PLTA)

Pembangkit Listrik Tenaga Air (PLTA) adalah salah satu pembangkit yang memanfaatkan aliran air untuk diubah menjadi energi listrik. Pembangkit listrik ini bekerja dengan cara merubah energi air yang mengalir (dari bendungan atau air terjun) menjadi energi mekanik (dengan bantuan turbin air) dan dari energi mekanik menjadi energi listrik (dengan bantuan generator). Kemudian energi listrik tersebut dialirkan melalui jaringan-jaringan yang telah dibuat, hingga akhirnya energi listrik tersebut sampai ke rumah kita.

Mesin penggerak yang digunakan adalah turbin air untuk mengubah energi potensial air menjadi kerja mekanis poros yang akan memutar rotor generator untuk menghasilkan energi listrik. Air sebagai bahan baku PLTA dapat diperoleh dari sungai secara langsung disalurkan untuk memutar turbin, atau dengan cara ditampung dahulu (bersamaan dengan air hujan) dengan menggunakan kolam tandon atau waduk sebelum disalurkan untuk memutar turbin.

2. Pembangkit Listrik Tenaga Matahari (*Photovoltaic* / PV)

Pembangkit listrik tenaga surya itu konsepnya sederhana. Yaitu mengubah cahaya matahari menjadi energi listrik. Cahaya matahari merupakan salah satu bentuk energi dari sumber daya alam. Sumber daya alam matahari ini sudah banyak digunakan untuk memasok daya listrik di satelit komunikasi melalui sel surya. Sel surya ini dapat menghasilkan energi listrik dalam jumlah yang tidak terbatas langsung diambil dari matahari, tanpa ada bagian yang berputar dan tidak memerlukan bahan bakar. Sehingga sistem sel surya sering dikatakan bersih dan ramah lingkungan.

Sistem sel surya yang digunakan di permukaan bumi terdiri dari panel sel surya, rangkaian kontroler pengisian (*charge controller*), dan aki (baterai) 12 volt yang *maintenance free*. Panel sel surya merupakan modul yang terdiri beberapa sel surya yang digabung dalam hubungan seri dan paralel tergantung ukuran dan kapasitas yang diperlukan. Yang sering digunakan adalah modul sel surya 20 watt atau 30 watt. Modul sel surya itu menghasilkan energi listrik yang proporsional dengan luas permukaan panel yang terkena sinar matahari.

3. Pembangkit Listrik Tenaga Angin

Pembangkit listrik tenaga angin adalah suatu pembangkit listrik yang menggunakan angin sebagai sumber energi untuk menghasilkan energi listrik. Pembangkit ini mengkonversikan energi angin menjadi energi listrik dengan menggunakan turbin angin atau kincir angin. Sistem pembangkitan listrik menggunakan angin sebagai sumber energi merupakan sistem alternatif yang sangat berkembang pesat, mengingat angin merupakan salah satu energi yang tidak terbatas di alam.

Suatu pembangkit listrik dari energi angin merupakan hasil dari penggabungan dari beberapa turbin angin sehingga akhirnya dapat menghasilkan listrik. Cara kerja dari pembangkitan listrik tenaga angin ini yaitu awalnya energi angin memutar turbin angin. Turbin angin bekerja berkebalikan dengan kipas angin (bukan menggunakan listrik untuk menghasilkan listrik, namun menggunakan angin untuk menghasilkan listrik). Kemudian angin akan memutar sudut turbin, lalu diteruskan untuk memutar rotor pada generator di bagian belakang turbin angin. Generator mengubah energi gerak menjadi energi listrik.

Berikut ini contoh pembangkit dengan sumber energi fosil:

1. Pembangkit Listrik Tenaga Uap

Pembangkit Listrik Tenaga Uap adalah pembangkit yang mengandalkan energi kinetik dari uap untuk menghasilkan energi listrik. Bentuk utama pembangkit listrik jenis ini adalah Generator yang di hubungkan ke turbin dimana untuk memutar turbin diperlukan energi kinetik dari uap panas atau kering. Pembangkit listrik tenaga uap menggunakan berbagai macam bahan bakar terutama batu-bara dan minyak bakar serta MFO untuk start awal. cara kerja PLTU sangat sederhana yaitu bahan bakar batubara dibuat untuk memanaskan *boiler*. Dalam *boiler* ini terdapat pipa – pipa yang berisi air. Semakin lama air dalam pipa tersebut mengalami penguapan dan kenaikan suhu yang signifikan. Sehingga uap air memiliki tekanan yang tinggi dan memutar turbin gas. Putaran turbin yang satu *shaft* dengan generator kemudian menghasilkan energi listrik. Uap air yang bersuhu tinggi tadi kemudian didinginkan melalui kondensor. Pendingin kondensor sendiri biasanya memanfaatkan air laut.

2. Pembangkit Listrik Tenaga Gas (PLTG/PLTGU)

Pusat listrik tenaga gas (PLTG) merupakan sebuah pembangkit energi listrik yang menggunakan peralatan/mesin turbin gas sebagai penggerak generatornya. Turbin gas dirancang dan dibuat dengan prinsip kerja yang sederhana dimana energi panas yang dihasilkan dari proses pembakaran bahan bakar diubah menjadi energi mekanis dan selanjutnya diubah menjadi energi listrik atau energi lainnya sesuai dengan kebutuhannya. Adapun kekurangan dari turbin gas adalah sifat korosif pada material yang digunakan untuk komponen-komponen turbinnya karena harus bekerja pada temperature tinggi dan adanya unsur kimia bahan bakar minyak yang korosif (sulfur, vanadium dll), tetapi dalam perkembangannya pengetahuan material yang terus berkembang hal tersebut mulai dapat dikurangi meskipun tidak dapat secara keseluruhan dihilangkan. Dengan tingkat efisiensi yang rendah hal ini merupakan salah satu dari kekurangan sebuah turbin gas juga dan pada perkembangannya untuk menaikkan efisiensi dapat diatur/diperbaiki temperature kerja siklus dengan menggunakan material turbin yang mampu bekerja pada temperature tinggi dan dapat juga untuk menaikkan efisiensinya dengan menggabungkan antara pembangkit turbin gas dengan pembangkit turbin uap dan hal ini biasa disebut dengan *combined cycle*.

2. 1. 2. Transmisi dan Distribusi jaringan listrik

Untuk mentransmisikan Listrik dari pembangkit menuju pelanggan diperlukan sebuah saluran transmisi. Transmisi dibagi menjadi dua jenis penyaluran yaitu saluran atas tanah dan saluran bawah tanah. Di Indonesia jalur transmisi yang digunakan hampir seluruhnya menggunakan jalur di atas tanah (*overhead*). Karena jalur transmisi atas tanah dinilai lebih mudah perawatannya dan harganya yang lebih terjangkau. Saluran transmisi dibagi menjadi beberapa jenis tegangan sesuai dengan jarak dari pembangkit sampai ke pelanggan. Semakin jauh jarak dari pembangkit sampai ke pelanggan digunakan tegangan yang lebih tinggi untuk menekan rugi daya yang terdapat pada saluran. Berdasarkan tegangannya saluran transmisi di Indonesia dibagi menjadi :

1. SUTR (40 – 1000 V)

Saluran ini disebut juga dengan saluran distribusi, yakni hilir dari saluran transmisi itu sendiri. Dengan kata lain, energi listrik yang dihasilkan oleh generator digunakan untuk memasok pelanggan kebutuhan konsumen rumah tangga. Di Indonesia saluran ini memiliki tegangan operasi 220/380V yang berarti tegangan bernilai 220 Volt *line to netral* dan 380 Volt *line to line*.

2. SUTM (6 – 30 KV)

Disebut juga saluran udara tegangan menengah (SUTM), dengan tegangan operasi antara 6 KV hingga 30 KV. Di Indonesia tegangan 6 KV dan 30 KV mulai banyak ditinggalkan dan lebih condong menggunakan tegangan operasi 20 KV. Saluran tegangan menengah biasa digunakan pada hubungan antara Gardu Induk, penyulang (*feeder*), gardu distribusi, hingga konsumen industri atau pelanggan dengan kapasitas besar.

3. SUTT (70 – 150 KV)

Disebut juga saluran udara tegangan tinggi (SUTT). Saluran transmisi ini memiliki tegangan antara 70 KV hingga 150 KV. Namun untuk saat ini tegangan operasi 70 KV mulai banyak ditinggalkan dan lebih sering digunakan tegangan operasi 150 KV. Konfigurasi jaringan pada umumnya memiliki konfigurasi *single* atau *double circuit*. Dalam

single circuit terdapat 3 fasa saluran dengan 3 atau 4 kawat (termasuk netral). Untuk 3 kawat saluran netralnya biasanya menggunakan media tanah untuk saluran kembali. Apabila kapasitas daya yang disalurkan cukup besar bisa menggunakan lebih dari satu kawat, bisa dua atau empat kawat yang disebut dengan *bundle conductor*. Saluran ini efektif digunakan dengan jarak dibawah 100 km. Apabila lebih jauh nilai drop tegangan menjadi semakin besar.

4. SUTET (200 – 500 KV)

Disebut juga saluran udara tegangan ekstra tinggi (SUTET). Saluran transmisi ini memiliki tegangan operasi sebesar 200 KV hingga 750 KV. Untuk saat ini saluran ini merupakan saluran dengan tegangan paling tinggi di Indonesia dengan tegangan 500 KV. Pembangkitan tegangan yang sangat tinggi ini efektif digunakan untuk menyalurkan daya listrik yang jauh dari pembangkit dengan jarak diatas 100 km. Kapasitas dayanya juga sangat tinggi, yakni diatas 500 MW. Saat ini saluran 500 KV terdapat pada pulau jawa, yang membentang dari PLTU Paiton hingga Suralaya. Dari saluran ekstra tinggi ini sepanjang saluran terdapat beban maupun suplai yang telah terinterkoneksi antar pembangkit satu dengan lainnya.

2. 1. 3. Beban

Beban adalah merupakan akhir dari sistem aliran daya. Beban adalah penentu berapa daya yang harus dibangkitkan oleh sebuah pembangkit. Beban biasa disebut pengguna listrik, dimana beban terbagi menjadi dua jenis yaitu beban linier seperti mesin listrik, AC, kipas angin dll. Dan beban non linier seperti charger HP, Laptop, lampu LED dll.

Pada dasarnya beban terbesar berasal dari dunia industri, dimana semua peralatan membutuhkan daya aktif maupun reaktif yang sangat besar. Salah satu contohnya proses peleburan baja, pengelasan dan pencampuran bahan kimia. Oleh karena itu Industri skala menengah sampai besar biasanya mendapat suplai tegangan 20 KV bahkan samapai 150 KV.

2. 2. Analisis Aliran Daya

Analisis aliran daya adalah studi dasar untuk menganalisis sistem sistem tenaga listrik, dalam perencanaan ataupun operasi. Tujuan utama dari seluruh analisis aliran daya adalah menentukan besar dan sudut fasa

tegangan pada setiap bus, dengan diketahuinya tegangan maka daya aktif (P) dan daya reaktif (Q) dapat dihitung. Jika P dan Q pada dua buah bus diketahui maka aliran daya dengan jelas dapat diketahui, serta rugi-rugi daya saluran transmisi dapat diketahui.

Secara umum tujuan analisis aliran daya adalah sebagai berikut :

1. Untuk mengetahui tegangan dan sudut fasa masing-masing bus.
2. Untuk mengetahui kemampuan semua peralatan yang ada dalam sistem apakah dapat cukup untuk menyalurkan daya yang diinginkan.
3. Untuk memperoleh kondisi awal bagi studi-studi selanjutnya, yakni studi stabilitas, studi hubung singkat, dan studi rugi-rugi transmisi.

Analisis aliran daya pada sistem tenaga, bertujuan untuk melihat bagaimana aliran daya dalam sistem. Perubahan yang terlibat dalam perhitungan adalah daya. Dengan menggunakan daya sebagai variabel sebagai variabel dalam perhitungan, maka persamaan yang kita hadapi menjadi bukan persamaan linier. Sumber, merupakan sumber daya yang hanya boleh beroperasi pada batas daya dan tegangan tertentu. Sementara itu beban adalah bagian rangkaian yang menyerap daya, sehingga ia dapat dinyatakan sebagai besar daya yang diminta/diperlukan, pada tegangan tertentu. Suatu permintaan daya hanya dapat dilayani selama pembebanan tidak melampaui batas daya yang mampu disediakan oleh sumber. Jadi walaupun rangkaian tetap rangkaian linier, namun relasi daya antara sumber dan beban tidaklah linier. Oleh karena itu jika persamaan rangkaian dengan daya sebagai variabel merupakan persamaan nonlinier. Dalam memecahkan persamaan nonlinier ini kita memerlukan cara khusus.

Dalam analisis aliran daya, kita mengambil ketentuan-ketentuan sebagai berikut:

- a) Sistem dalam keadaan seimbang; dengan keadaan seimbang ini kita dapat melakukan perhitungan dengan menggunakan model satu-fasa.
- b) Semua besaran dinyatakan dalam per-unit; dengan menggunakan sistem satuan ini kita terbebas dari persoalan perbedaan tegangan di berbagai bagian sistem yang diakibatkan oleh pemanfaatan transformator dalam upaya penyaluran daya.

Terdapat tiga macam jenis bus dalam analisa aliran daya, sebagai berikut:

1. Bus referensi (*Slack bus*)

Slack bus (Bus referensi), yaitu suatu bus yang selalu mempunyai besaran dan sudut fasa yang tetap dan telah diberikan sebelumnya, pada bus ini berfungsi untuk mencatu rugi – rugi, kekurangan daya yang terdapat pada jaringan, dalam hal ini penting karena kekurangan daya tidak dapat dicapai kecuali terdapat suatu bus yang mempunyai daya yang tak terbatas sehingga dapat mengimbangi rugi - rugi

2. Bus Generator (*Voltage controlled bus*)

Pada tipe bus ini, besar tegangan daya aktif telah ditentukan sedangkan daya reaktif dan sudut fasa tegangan di dapat dari hasil perhitungan.

3. Bus beban (*Load bus*)

Pada tipe bus ini daya aktif dan daya reaktif diketahui, sedangkan dua lainnya didapat dari hasil perhitungan.

Adapun untuk studi analisa aliran daya, data – data yang dibutuhkan dari bus – bus dan saluran transmisi, yaitu :

1. *Magnitude* tegangan (dalam satuan p.u.)
2. Sudut tegangan (θ)
3. *Constraint* pembangkitan, meliputi daya aktif (Pmin dan Pmax) dan daya reaktif (Qmin dan Qmax)
4. Daya pembangkitan meliputi daya aktif (P) dan daya reaktif (Q)
5. Daya pembebanan meliputi daya aktif (P) dan daya reaktif (Q)

2. 3. Persamaan Aliran Daya

Persamaan aliran daya merupakan persamaan matematis yang digunakan dalam menyelesaikan persoalan aliran daya pada sebuah sistem.

Admitansi – admitansi pada sistem tenaga listrik diatas, dijadikan dalam satuan per unit (pu) pada dasar / base MVA. Persamaan (2.1) untuk mencari nilai dari arus :

$$I_{Base} = \frac{KVA_{Base}-1\phi}{KV_{Base}-LN} \quad (2.1)$$

Persamaan (2.2) untuk mencari nilai impedansi dasar :

$$Z_{Base} = \frac{(KV_{Base}-LN)^2 \times 1000}{KVA_{Base}-1\phi} \quad (2.2)$$

Matriks admitansi jaringan di :

$$= \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & Y_{23} & \dots & Y_{2n} \\ Y_{31} & Y_{32} & Y_{33} & \dots & Y_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & Y_{n3} & \dots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \dots \\ V_n \end{bmatrix} \quad (2.3)$$

Keterangan konstanta yang digunakan pada persamaan (2.3) :

- n : Jumlah total *node*
- Y_{ii} : Admitansi dari *node* i
- Y_{ij} : Admitansi dari *node* i dan j
- V_i : Tegangan fasor pada *node* i
- I_i : Arus fasor yang mengalir ke *node* i

Dengan menguraikan persamaan (2.3) dengan menggunakan hukum Kirchoff, maka akan didapatkan persamaan (2.4) dan persamaan (2.5) sebagai berikut :

$$I_i = y_{i0} V_i + y_{i1} (V_i - V_1) + y_{i2} (V_i - V_2) + y_{i3} (V_i - V_3) + \dots + y_{in} (V_i - V_n) \quad (2.4)$$

$$I_i = (y_{i0} + y_{i1} + y_{i2} + y_{i3} + \dots + y_{in}) V_i - y_{i1} V_1 - y_{i2} V_2 - y_{i3} V_3 - \dots - y_{in} V_n \quad (2.5)$$

Persamaan (2.5) dapat ditulis menjadi :

$$I_i = V_i \sum_{j=0}^n Y_{ij} - \sum_{j=1}^n Y_{ij} V_j, j \neq 1 \quad (2.6)$$

Persamaan untuk arus dan tegangan dalam bentuk polar ditulis pada persamaan (2.7) dan persamaan (2.8) sebagai berikut :

$$V = |V| \angle \alpha \quad (2.7)$$

$$I = |I| \angle \beta \quad (2.8)$$

Persamaan daya kompleks dituliskan pada persamaan (2.9) sebagai berikut :

$$S = V I^* \quad (2.9)$$

Keterangan konstanta yang digunakan pada persamaan (2.7), persamaan (2.8), dan persamaan (2.9) sebagai berikut :

- V : Besar tegangan
- I : Besar arus
- α : Besar sudut dari tegangan
- β : Besar sudut dari arus
- S : Daya kompleks

Persamaan (2.7) dan persamaan (2.8) disubstitusi ke persamaan (2.9) akan menjadi sebagai berikut :

$$S = |V| \angle \alpha (|I| \angle \beta)^* \quad (2.10)$$

$$S = |V| \angle \alpha |I| \angle -\beta \quad (2.11)$$

$$S = |V| |I| \angle (\alpha - \beta) \quad (2.12)$$

$$S = |V| |I| \angle \theta \quad (2.13)$$

Persamaan (2.12) dan persamaan (2.13) dijadikan dalam bentuk persamaan *rectangular* pada persamaan (2.14) dan persamaan (2.15) sebagai berikut :

$$S = |V| |I| \cos(\alpha - \beta) + j |V| |I| \sin(\alpha - \beta) \quad (2.14)$$

$$S = |V| |I| \cos(\theta) + j |V| |I| \sin(\theta) \quad (2.15)$$

Dengan memisahkan bagian real dan imajiner pada persamaan (2.15), maka akan didapatkan persamaan daya aktif dan persamaan daya reaktif sebagai berikut :

$$P = |V| |I| \cos(\theta) \quad (2.16)$$

$$Q = j |V| |I| \sin(\theta) \quad (2.17)$$

2.3.1. Beban induktif (*Lagging*)

Beban induktif yaitu beban yang terdiri dari kumparan kawat yang dililitkan pada suatu inti, seperti coil, transformator, dan solenoida. Beban ini dapat menghalangi perubahan nilai arus, mengakibatkan fasa arus bergeser menjadi tertinggal terhadap tegangan - pergeseran fasa (*phase shift*) pada beban induktif ini biasa disebut *lagging* - ,hal ini disebabkan oleh energi yang tersimpan berupa medan magnetis. Beban jenis ini menyerap daya aktif dan daya reaktif.

Tegangan V ditetapkan sebagai referensi terhadap arus I , sehingga dapat diturunkan dari persamaan matematis (2.18) dan persamaan (2.19) sebagai berikut :

$$V = |V| \angle 0 \quad (2.18)$$

$$I = |I| \angle (-\phi) \quad (2.19)$$

Dengan mensubstitusi persamaan (2.18) dan persamaan (2.19) ke persamaan (2.9), maka didapatkan persamaan (2.20) sebagai berikut :

$$S = |V| \angle 0 (|I| \angle -\phi)^* \quad (2.20)$$

$$S = |V| |I| \angle (\phi + 0) \quad (2.21)$$

$$S = |V| |I| \angle \phi \quad (2.22)$$

Persamaan daya dalam kondisi *lagging* didapatkan dengan merubah persamaan (2.22) kedalam bentuk *rectangular* sebagai berikut :

$$S = |V| |I| \cos(\phi) + j |V| |I| \sin(\phi) \quad (2.23)$$

$$S = P + j Q \quad (2.24)$$

Persamaan (2.23) dan persamaan (2.24) merupakan representasi dari persamaan daya kompleks dalam kondisi *lagging*.

2.3.2. Beban kapasitif (*Leading*)

Beban kapasitif adalah kebalikan dari beban induktif. Jika beban induktif dapat menghalangi terjadinya perubahan nilai arus, maka beban kapasitif bersifat menghalangi perubahan nilai tegangan. Sifat inilah yang menyebabkan kapasitor seakan-akan menyimpan tegangan listrik sesaat. Beban ini dapat mengakibatkan fasa arus bergeser menjadi mendahului terhadap tegangan - pergeseran fasa (phase shift) pada beban kapasitif ini biasa disebut *leading*. Beban jenis ini menyerap daya aktif dan mengeluarkan daya reaktif.

Pada gambar (2.4), arus I ditetapkan sebagai referensi terhadap tegangan V , sehingga dapat diturunkan persamaan matematis (2.25) dan persamaan (2.26) sebagai berikut :

$$V = |V| \angle 0 \quad (2.25)$$

$$I = |I| \angle 0 \quad (2.26)$$

Persamaan daya kompleks dinyatakan dalam persamaan (2.27) berikut :

$$S = V I \quad (2.27)$$

Dengan mensubstitusi persamaan (2.25) dan persamaan (2.26) ke persamaan (2.27), maka didapatkan persamaan (2.28) berikut :

$$S = (|V| \angle 0) * |I| \angle 0 \quad (2.28)$$

$$S = |V| \angle (-0) |I| \angle 0 \quad (2.29)$$

$$S = |V| |I| \angle (-0 + 0) \quad (2.30)$$

$$S = |V| |I| \angle - 0 \quad (2.31)$$

Persamaan daya dalam kondisi *leading* didapatkan dengan merubah persamaan (2.31) kedalam bentuk *rectangular* sebagai berikut :

$$S = |V| |I| \cos(0) - j |V| |I| \sin(0) \quad (2.32)$$

$$S = P - j Q \quad (2.33)$$

Persamaan (2.31) dan persamaan (2.32) merupakan representasi dari persamaan daya kompleks dalam kondisi *leading*.

2.3.3. Penyelesaian Aliran Daya Metode Newton Rhapson

Bentuk matematis arus pada bus admitansi dituliskan pada persamaan (2.34) sebagai berikut :

$$I_i = \sum_{j=1}^n Y_{ij} V_j \quad (2.34)$$

Persamaan (2.34) dalam bentuk *polar* diberikan dalam persamaan (2.35) berikut :

$$I_i = \sum_{j=1}^n Y_{ij} V_j \angle \theta_{ij} + \delta_j \quad (2.35)$$

Perhitungan daya kompleks pada bus i menggunakan persamaan (2.36) sebagai berikut :

$$P_i - jQ_i = V_i * I_i \quad (2.36)$$

Dengan mensubstitusi persamaan (2.35) ke (2.36), maka akan didapatkan hasil sebagai berikut :

$$P_i - jQ_i = V_i * \sum_{j=1}^n Y_{ij} V_j \angle \theta_{ij} + \delta_j \quad (2.37)$$

$$P_i - jQ_i = |V| \angle -\delta_i \sum_{j=1}^n Y_{ij} V_j \angle \theta_{ij} + \delta_j \quad (2.38)$$

$$P_i - jQ_i = |V| \sum_{j=1}^n Y_{ij} V_j \angle \theta_{ij} + \delta_j - \delta_i \quad (2.39)$$

Persamaan (2.39) dalam bentuk *rectangular* diberikan pada persamaan (2.40) sebagai berikut :

$$P_i - jQ_i = |V| \sum_{j=1}^n Y_{ij} V_j \cos(\theta_{ij} + \delta_j - \delta_i) - j |V| \sum_{j=1}^n Y_{ij} V_j \sin(\theta_{ij} + \delta_j - \delta_i) \quad (2.40)$$

Dengan memisahkan bagian *real* dan *imaginer* persamaan (2.40) didapatkan rumus sebagai berikut :

$$P_i = |V| \sum_{j=1}^n Y_{ij} V_j \cos(\theta_{ij} + \delta_j - \delta_i) \quad (2.41)$$

$$Q_i = -|V| \sum_{j=1}^n Y_{ij} V_j \sin(\theta_{ij} + \delta_j - \delta_i) \quad (2.42)$$

Persamaan (2.40) dan persamaan (2.41) merupakan persamaan nonlinier, yang memiliki besaran tegangan dalam satuan per unit (pu) dan sudut dalam radian.

Dengan menggunakan deret Taylor, maka akan didapatkan rumus sebagai berikut :

$$\begin{bmatrix} \Delta P_2^{(k)} \\ \dots \\ \Delta P_n^{(k)} \\ \Delta Q_2^{(k)} \\ \dots \\ \Delta Q_n^{(k)} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2^{(k)}}{\partial \delta_2} & \dots & \frac{\partial P_2^{(k)}}{\partial \delta_n} & \frac{\partial P_2^{(k)}}{\partial |V_2|} & \dots & \frac{\partial P_2^{(k)}}{\partial |V_n|} \\ \frac{\partial P_n^{(k)}}{\partial \delta_2} & \dots & \frac{\partial P_n^{(k)}}{\partial \delta_n} & \frac{\partial P_n^{(k)}}{\partial |V_2|} & \dots & \frac{\partial P_n^{(k)}}{\partial |V_n|} \\ \frac{\partial Q_2^{(k)}}{\partial \delta_2} & \dots & \frac{\partial Q_2^{(k)}}{\partial \delta_n} & \frac{\partial Q_2^{(k)}}{\partial |V_2|} & \dots & \frac{\partial Q_2^{(k)}}{\partial |V_n|} \\ \frac{\partial Q_n^{(k)}}{\partial \delta_2} & \dots & \frac{\partial Q_n^{(k)}}{\partial \delta_n} & \frac{\partial Q_n^{(k)}}{\partial |V_2|} & \dots & \frac{\partial Q_n^{(k)}}{\partial |V_n|} \end{bmatrix} \begin{bmatrix} \Delta \delta_2^{(k)} \\ \dots \\ \Delta \delta_n^{(k)} \\ \Delta |V_2^{(k)}| \\ \dots \\ \Delta |V_n^{(k)}| \end{bmatrix} \quad (2.43)$$

Proses differensiasi dari persamaan (2.43) diberikan pada persamaan (2.44) sampai dengan persamaan (2.53). Differensiasi dari P_i terhadap δ_i ditunjukkan pada persamaan (2.44) dan persamaan (2.45) sebagai berikut:

$$\frac{\partial P_i}{\partial \delta_i} = \frac{\sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \cos(\theta_{ij} + \delta_j - \delta_i)}{\partial \delta_i} \quad (2.44)$$

$$\frac{\partial P_i}{\partial \delta_i} = - \sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \sin(\theta_{ij} + \delta_j - \delta_i) \quad (2.45)$$

Differensiasi dari P_i terhadap V_i ditunjukkan pada persamaan (2.46), persamaan (2.47) dan persamaan (2.48) sebagai berikut :

$$\frac{\partial P_i}{\partial V_i} = \frac{\sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \cos(\theta_{ij} + \delta_j - \delta_i)}{\partial V_i} \quad (2.46)$$

$$\frac{\partial P_i}{\partial V_i} = \frac{\sum_{j=1}^n |Y_{ij}| |V_i|^2 \cos(\theta_{ij} + \delta_j - \delta_i)}{\partial V_i} \quad (2.47)$$

$$\frac{\partial P_i}{\partial \delta_i} = 2 \sum_{j=1}^n |Y_{ij}| |V_i| \cos(\theta_{ij} + \delta_j - \delta_i) \quad (2.48)$$

Differensiasi dari Q_i terhadap δ_i ditunjukkan pada persamaan (2.49) dan persamaan (2.50) sebagai berikut :

$$\frac{\partial Q_i}{\partial \delta_i} = \frac{-\sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \sin(\theta_{ij} + \delta_j - \delta_i)}{\partial \delta_i} \quad (2.49)$$

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \cos(\theta_{ij} + \delta_j - \delta_i) \quad (2.50)$$

Differensiasi dari Q_i terhadap δ_i ditunjukkan pada persamaan (2.51), persamaan (2.52) dan persamaan (2.53) sebagai berikut :

$$\frac{\partial Q_i}{\partial V_i} = \frac{-\sum_{j=1}^n |Y_{ij}| |V_j| |V_i| \sin(\theta_{ij} + \delta_j - \delta_i)}{\partial V_i} \quad (2.51)$$

$$\frac{\partial Q_i}{\partial V_i} = \frac{-\sum_{j=1}^n |Y_{ij}| |V_i|^2 \sin(\theta_{ij} + \delta_j - \delta_i)}{\partial V_i} \quad (2.52)$$

$$\frac{\partial Q_i}{\partial V_i} = -2 \sum_{j=1}^n |Y_{ij}| |V_i| \sin(\theta_{ij} + \delta_j - \delta_i) \quad (2.53)$$

Hasil dari proses differerensiasi pada persamaan (2.45), persamaan (2.48), persamaan (2.50), dan persamaan (2.53) akan dipergunakan pada perhitungan aliran daya metode Newton Rhapson.

2. 4. Optimal Power Flow

Optimal Power Flow (OPF) pertama kali dibahas oleh Carpentier pada tahun 1962 dan membutuhkan waktu yang sangat lama untuk membentuk penyelesaian algoritma yang dapat diaplikasikan pada saat ini.

OPF memiliki tujuan untuk menentukan pengaturan yang optimal pada jaringan sistem tenaga listrik dengan mengoptimisasi fungsi – fungsi obyektif seperti fungsi total biaya pembangkitan (*Economic Dispatch*), fungsi rugi – rugi (*losses*), fungsi emisi pembangkitan dengan mempertimbangkan batasan operasi peralatan.

Secara umum, fungsi OPF dapat dituliskan dalam persamaan (2.54) sebagai berikut :

$$\text{Min } f(x, u)$$

$$\text{Subject to : } g(x, u) \quad (2.54)$$

$$h(x, u) \leq 0$$

$$x = [\theta, V_L, P_{g \text{ slack}}, Q_g]^T \quad (2.55)$$

$$u = [P_g, V_g, t, B_{sh}]^T \quad (2.56)$$

Keterangan parameter pada persamaan (2.54), (2.55) dan persamaan (2.56) sebagai berikut :

$f(x, u)$: Fungsi obyektif.
$g(x, u)$: Representasikan persamaan daya dan reaktif.
$b(x, u)$: <i>Security limit</i> .
x	: Vektor variabel dependen.
u	: Vektor variabel independen.
V_L	: <i>Voltage magnitude</i> dari bus beban.
P_g	: Daya aktif pembangkitan.
$P_{g \text{ slack}}$: Daya aktif pembangkitan.
Q_g	: Daya reaktif pembangkitan.
V_g	: Tegangan pada bus generator.
t	: Rasio tap transformator.
V_g	: Tegangan pada bus generator.

2.4.1 Fungsi Active Power Dispatch

Pada multiobjective active power dispatch terdapat fungsi obyektif yang digunakan sebagai berikut :

- Fungsi *Operation Cost* (Biaya Pembangkitan)
- Fungsi *Emission* (Emisi)

2.4.1.1 Fungsi Operation Cost (Biaya Pembangkitan)

Fungsi biaya pembangkitan yang minimum diberikan pada persamaan (2.57) dan persamaan (2.58) sebagai berikut :

$$F = \sum_{i=1}^{NG} F_i(P_i) \quad (2.57)$$

$$F_c(P_g) = \sum_{i=1}^{Ng} a_i + b_i P_{gi} + C_i P_{gi}^2 \quad (2.58)$$

Keterangan parameter persamaan (2.57) dan persamaan (2.58) sebagai berikut :

- F : Nilai total biaya pembangkitan.
 $F_i(P_i)$: Nilai biaya pembangkitan dari pembangkit ke – i yang merupakan fungsi dari output daya pembangkitan.
 NG : Nilai jumlah dari unit pembangkit.
 a_i, b_i, c_i : Nilai koefisien biaya pembangkit.

Fungsi obyektif untuk emisi dari pembangkit diberikan persamaan (2.59)

$$F_e(P_g) = \sum_{i=1}^{NG} 10^{-2}(a_i + b_i P_i + c_i P_i^2) + C_i e^{P_{gia}} \quad (2.59)$$

2.4.1.2 Fungsi *Carbon Emission* (Emisi Karbon)

Jumlah total emisi polutan dari unit pembangkit termal fosil tergantung pada jumlah daya yang dihasilkan oleh masing-masing unit . Untuk penyederhanaan, emisi total yang dihasilkan dapat dimodelkan sebagai fungsi kuadrat dan istilah eksponensial output daya aktif dari unit pembangkit. Emisi polutan dapat digambarkan sebagai optimalisasi jumlah emisi polutan didefinisikan oleh persamaan berikut:

Fungsi emisi karbon yang minimum diberikan pada persamaan (2.58) dan persamaan (2.59) sebagai berikut :

$$Er = \sum_{i=1}^{NG} F_i(P_i) \quad (2.60)$$

$$Er(P_i) = \sum_{i=1}^{NG} \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \varepsilon_i e^t \quad (2.61)$$

Keterangan parameter persamaan (2.58) dan persamaan (2.62) sebagai berikut :

- E : Nilai total emisi karbon
 $E_r(P_i)$: Nilai emisi karbon dari pembangkit ke – i yang merupakan fungsi dari output daya pembangkitan.
 NG : Nilai jumlah dari unit pembangkit.
 $\alpha_i, \beta_i, \gamma_i$: Nilai koefisien emisi karbon pembangkit ke - i.

Bila suatu unit pembangkit dioperasikan atau dibebani diluar batas maksimum dan minimumnya selain efisiensinya rendah, umur

(*lifetime*) dari mesin tersebut akan menurun terutama bila sering mengalami pembebanan lebih (*overloading*). Oleh karena itu agar pembangkit tersebut selalu dapat bekerja dengan efisiensi yang cukup baik (ekonomis) serta stabil, maka pembangkit tersebut harus dioperasikan dalam daerah pembebanannya.

Emission dispatch adalah pembagian pembebanan pada setiap unit pembangkit sehingga diperoleh kombinasi unit pembangkit yang dapat memenuhi kebutuhan beban dengan total emisi karbon yang optimum atau dengan kata lain untuk mencari nilai optimum dari output daya dari kombinasi unit pembangkit yang bertujuan untuk meminimalkan total emisi karbon pembangkitan.

2. 5. Aplikasi Optimal Power Flow

Optimal Power Flow memberikan banyak kegunaan di bidang aplikasi sebagai berikut :

1. Memberikan perhitungan pembangkitan yang optimum dengan banyak variabel yang dikontrol, yang bertujuan meminimalkan biaya pembangkitan dengan memperhitungkan limit dari sistem transmisi.
2. OPF mampu digunakan untuk menemukan pengaturan tegangan pembangkitan dan tap transformator yang optimum

2. 6. Particle Swarm Optimization (PSO)

Particle Swarm Optimization adalah salah satu metode optimasi yang terinspirasi dari perilaku gerakan kawanan hewan seperti ikan (*school of fish*), hewan *herbivor* (*herd*), dan burung (*flock*) yang kemudian tiap objek hewan disederhanakan menjadi sebuah partikel. Suatu partikel dalam ruang memiliki posisi yang dikodekan sebagai vektor koordinat. Vektor posisi ini dianggap sebagai keadaan yang sedang ditempati oleh suatu partikel di ruang pencarian. Setiap posisi dalam ruang pencarian merupakan alternatif solusi yang dapat dievaluasi menggunakan fungsi objektif. Setiap partikel bergerak dengan kecepatan v .

Particle Swarm Optimization atau yang kita kenal dengan PSO menerapkan sifat masing-masing individu dalam satu kelompok besar. Kemudian menggabungkan sifat-sifat tersebut untuk menyelesaikan permasalahan. Particle Swarm Optimization pertama kali dimunculkan pada tahun 1995, sejak saat itulah para peneliti banyak menurunkan dan mengembangkan metode PSO.

Ciri khas dari PSO adalah pengaturan kecepatan partikel secara heuristik dan probabilistik. Jika suatu partikel memiliki kecepatan yang konstan maka jika jejak posisi suatu partikel divisualisasikan akan membentuk garis lurus. Dengan adanya faktor eksternal yang membelokkan garis tersebut yang kemudian menggerakkan partikel dalam ruang pencarian maka diharapkan partikel dapat mengarah, mendekati, dan pada akhirnya mencapai titik optimal. Faktor eksternal yang dimaksud antara lain posisi terbaik yang pernah dikunjungi suatu partikel, posisi terbaik seluruh partikel (diasumsikan setiap partikel mengetahui posisi terbaik setiap partikel lainnya), serta faktor kreativitas untuk melakukan eksplorasi.

Particle Swarm Optimization memiliki kesamaan sifat dengan teknik komputasi seperti Algoritma Genetika (Genetic Algorithm). Sistem PSO diinisialisasi oleh sebuah populasi solusi secara acak dan selanjutnya mencari titik optimum dengan cara meng-update tiap hasil pembangkitan. Metode optimasi yang didasarkan pada swarm intelligence ini disebut algoritma behaviorally inspired sebagai alternatif dari algoritma genetika, yang sering disebut evolution-based procedures. Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik: posisi dan kecepatan. Setiap partikel bergerak dalam ruang/space tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut.

Sebagai contoh, misalnya perilaku burung-burung dalam dalam kawanan burung. Meskipun setiap burung mempunyai keterbatasan dalam hal kecerdasan, biasanya ia akan mengikuti kebiasaan (rule) seperti berikut :

1. Seekor burung tidak berada terlalu dekat dengan burung yang lain.
2. Burung tersebut akan mengarahkan terbangnya ke arah rata-rata keseluruhan burung.
3. Burung akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh.

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor simpel berikut:

1. Kohesi - terbang bersama
2. Separasi - jangan terlalu dekat
3. Penyesuaian (alignment) - mengikuti arah bersama

Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu :

$$X_{it}(t) = x_{i1}(t), x_{i2}(t), \dots, x_{iN}(t) \quad (2.63)$$

$$V_i(t) = v_{i1}(t), v_{i2}(t), \dots, v_{iN}(t) \quad (2.64)$$

Keterangan persamaan (2.63) dan (2.64) sebagai berikut :

- | | |
|---|------------------------|
| X | : Posisi partikel |
| V | : Kecepatan partikel |
| I | : Indeks partikel |
| T | : Iterasi ke-t |
| N | : Ukuran dimensi ruang |

Berikut ini merupakan model matematika yang menggambarkan mekanisme *updating status* partikel Kennedy and Eberhart [1995]:

$$\begin{aligned} V_i(t) &= V_i(t-1) + c1r1(X_i^L - X_i(t-1)) + c2r2(X^G - X_i(t-1)) \\ X_i(t) &= X_i(t-1) + V_i(t) \end{aligned} \quad (2.65)$$

Dimana :

$$X_i^L = X_{i1}^L, X_{i2}^L, \dots, X_{iN}^L \quad (2.66)$$

Merepresentasikan local best dari partikel ke-i. Sedangkan, $X_i^G = X_1^G, X_2^G, \dots, X_N^G$ merepresentasikan global best dari seluruh kawanan. Sedangkan c1 dan c2 adalah suatu konstanta yang bernilai positif yang biasanya disebut sebagai *learning factor*. Kemudian r1 dan r2 adalah suatu bilangan *random* yang bernilai antara 0 sampai 1, digunakan untuk menghitung kecepatan partikel yang baru berdasarkan kecepatan sebelumnya, jarak antara posisi saat ini dengan posisi terbaik partikel

(*local best*), dan jarak antara posisi saat ini dengan posisi terbaik kawanannya (*global best*). Kemudian partikel terbang menuju posisi yang baru. Setelah algoritma PSO ini dijalankan dengan sejumlah iterasi tertentu hingga mencapai kriteria pemberhentian, maka akan didapatkan solusi yang terletak pada *global best*. Model ini akan disimulasikan dalam ruang dengan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi). Ini dilakukan hingga maksimum iterasi dicapai atau bisa juga digunakan kriteria penghentian yang lain.

2. 7. Cuckoo Search Algorithm

Algoritma burung Cuckoo atau *Cuckoo search* merupakan algoritma *metaheuristik* yang relatif baru dibandingkan dengan metode optimisasi lainnya. Walaupun Algoritma penelitian yang bekerja terkait dengan aplikasi Algoritma Cuckoo yang telah dilaporkan dalam berbagai literatur. Algoritma ini telah sepenuhnya diterapkan untuk menyelesaikan fungsi matematika terus menerus dan masalah pada industri.

Algoritma Cuckoo merupakan tipe algoritma *metaheuristik* baru yang diperkenalkan oleh Xin She Yang pada tahun 2010. Algoritma ini terinspirasi oleh perilaku burung cuckoo. Burung cuckoo adalah burung yang sangat kejam karena Burung cuckoo tidak pernah memiliki sarang dan keluarganya sendiri. Sebelum menetas telurnya, sang induk burung cuckoo akan mengincar sarang burung lain untuk menitipkan telurnya. Sang induk akan mencari telur yang hampir mirip dengan bentuk telurnya. Ketika burung yang lain itu kembali, dia akan mengira bahwa itu adalah telurnya juga. Sehingga dia akan mengerami bersamaan dengan telurnya sendiri. Perilaku parasit pada cuckoo dibagi menjadi tiga, pembiakan parasit intraspesifik, pembiakan kooperatif dan pengambil alihan sarang. Ada beberapa jenis cuckoo yang membuang telur dari induk asli pada sarang untuk meningkatkan kemungkinan menetas telur mereka. mungkin saja terjadi konflik antara burung host dan burung cuckoo pada saat burung cuckoo menaruh telurnya, sehingga burung host membuang telur cuckoo tersebut atau meninggalkan sarang mereka kemudian membuat sarang yang baru. Perilaku parasit yang lain adalah saat burung cuckoo menetas, biasanya telur burung cuckoo menetas lebih awal dari telur burung host, telur burung yang belum menetas tadi dibuang dari sarang agar anak cuckoo tersebut

mendapatkan lebih banyak makanan. Bahkan cuckoo mampu melakukan penyamaran pada suara burung host kecil.

Pada program, untuk penyederhanaan, penemu CS Xin She Yang dan Suash Deb menggunakan beberapa aturan antar lain :

1. Setiap cuckoo meletakkan satu telur pada satu waktu dan membuang telur di dalamnya pada sarang yang dipilih secara acak,

2. Sarang terbaik dengan kualitas telur tertinggi akan dibawa ke generasi berikutnya,

3. Jumlah sarang burung inang yang tersedia berjumlah tetap, dan peluang telur yang diletakkan oleh seekor cuckoo ditemukan burung inang sebesar $pa \in [0,1]$. Dalam kasus ini, burung inang dapat membuang telur cuckoo atau meninggalkan sarang, lalu membuat sebuah sarang baru.

Aturan terakhir dapat disederhanakan dengan pendekatan fraksi pa dari n sarang digantikan dengan sarang baru. Dalam *cuckoo search*, solusi direpresentasikan oleh satu telur dalam sarang dan solusi baru direpresentasikan sebagai telur. Kemudian telur-telur tadi akan dievolusikan, telur yang lebih baik akan menggantikan telur lainnya yang diukur oleh fitness, layaknya dalam GA.

Pseudo code dari Algoritma Cuckoo yang diusulkan oleh Yang pertama kali ditunjukkan sebagai berikut:

Fungsi objektif $f(X) = X(X_1, \dots, X_d)^T$
Inisialisasi populasi dari n sarang burung target Xi ($i = 1, 2 \dots n$);

while ($t < \text{Jumlah iterasi maksimum}$) atau (kriteria lain untuk berhenti);

Evaluasi nilai kualitas dari masing – masing burung cuckoo
Pilih burung cuckoo secara acak dan lakukan random walk

jika ($F_i > F_j$)

Gantikan burung cuckoo j dengan burung cuckoo i

End If

Reset ulang sarang-sarang dengan kondisi terburuk (Pa)
Simpan sarang-sarang yang berhasil lolos
Urutkan solusi dan cari yang terbaik

End While

2.7.1 Persamaan Pencarian Host

Pencarian sarang host baru atau inang dapat dituliskan dalam persamaan (2.71) :

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{L\'evy}(\lambda) \quad (2.65)$$

Keterangan persamaan (2.65) sebagai berikut :

$x_i^{(t+1)}$: Posisi Partikel pada iterasi n

$x_i^{(t)}$: Posisi Partikel awal

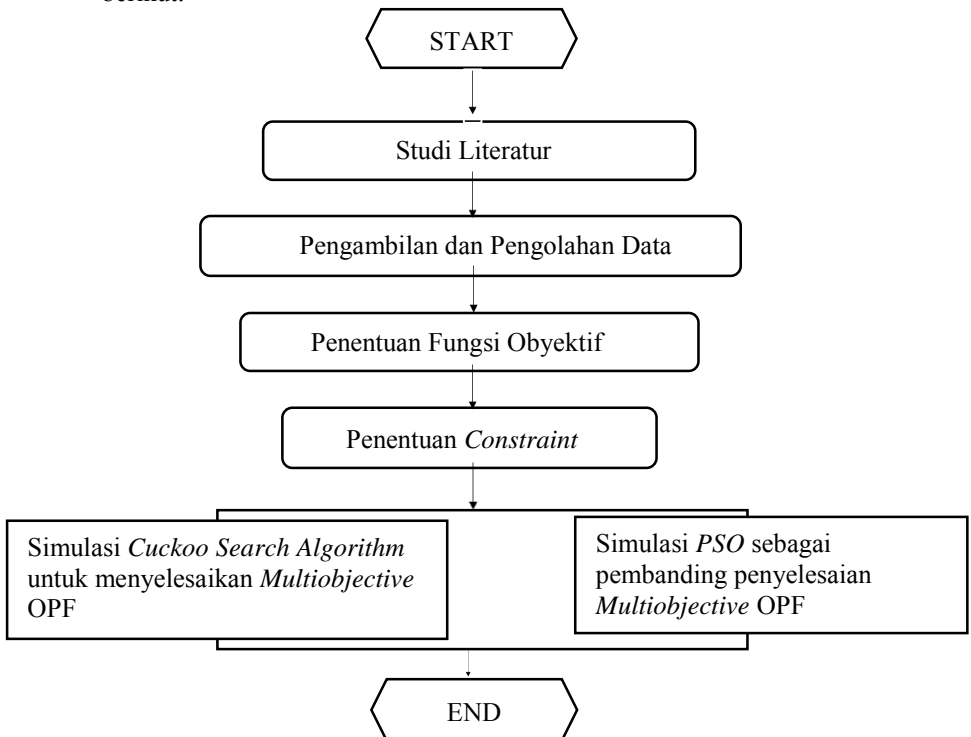
$\alpha \oplus \text{L\'evy}(\lambda)$: Konstanta levy flight

BAB 3

PENYELESAIAN MASALAH OPTIMASI DENGAN OPTIMAL POWER FLOW

3. 1. Perencanaan Penelitian

Dalam menyelesaikan penelitian *Optimal Power Flow* (OPF) terdapat beberapa tahapan yang dilalui. Dengan menggunakan metode *Cuckoo Search Algorithm* pada sistem tenaga listrik yang akan digambarkan pada diagram penelitian OPF pada gambar (3.1) sebagai berikut:



Gambar 3.1 : Flowchart tahap – tahap penelitian *Optimal Power Flow*

3. 2. Studi Literatur

Multiobjective Optimal Power Flow didefinisikan sebagai pengaturan yang optimal dari jaringan sistem tenaga listrik yang memenuhi nilai optimal dari beberapa fungsi obyektif. Studi literatur diperoleh dari jurnal – jurnal internasional dan nasional, proseding internasional dan nasional serta buku – buku teks yang relevan dengan bahasan penelitian.

1. Referensi [1, 2, 3, 4, 5] adalah buku teks yang berkaitan dengan Optimal Power Flow dan penyelesaian permasalahan Optimal Power Flow menggunakan metode heuristic
2. Referensi [6, 7, 8, 9] adalah jurnal dan prosiding yang berkaitan dengan Multiobjective Optimal Power Flow.
3. Referensi [6, 13, 14] adalah jurnal dan prosiding yang berkaitan dengan Particle Swarm Optimization
4. Referensi [10, 11, 12] adalah jurnal dan prosiding yang berkaitan dengan Cuckoo Search Algorithm.

3. 3. Pengolahan data

Data yang diperlukan untuk melakukan simulasi adalah data test standart IEEE 30 bus yang terdiri oleh :

1. Data daya pembangkitan
2. Data daya beban
3. Data impedansi saluran
4. Data karakteristik dispatch pembangkit
5. Data karakteristik emisi pembangkit

Data – data tersebut akan digunakan untuk *running load flow* dalam penentuan biaya pembangkitan yang paling optimal dan minimasi rugi – rugi pembangkitan sistem tenaga listrik

3. 4. Penentuan Fungsi Obyektif

Fungsi obyektif dalam permasalahan OPF bersifat statis dan nonlinear, dalam penelitian ini digunakan *multiobjective function* sebagai berikut :

3. 4. 1. Fungsi *Emision* (Emisi)

Fungsi emisi karbon yang minimum diberikan pada persamaan (3.1) dan persamaan (3.2) sebagai berikut :

$$Er = \sum_{i=1}^{NG} F_i(P_i) \quad (3.1)$$

$$Er(P_i) = \sum_{i=1}^{Ng} \alpha_i P_i^2 + \beta_i P_i + \gamma_i + \varepsilon_{i\epsilon t} \quad (3.2)$$

Keterangan parameter persamaan (3.1) dan persamaan (3.2) sebagai berikut :

E : Nilai total emisi karbon

$Er(P_i)$: Nilai emsisi karbon dari pembangkit ke – i yang merupakan fungsi dari output daya pembangkitan.

NG : Nilai jumlah dari unit pembangkit.

$\alpha_i, \beta_i, \gamma_i$: Nilai koefisien emisi larbon pembangkit ke - i.

3. 4. 2. Fungsi *Generation Cost* (Biaya Pembangkitan)

Fungsi biaya pembangkitan yang minimum diberikan pada persamaan (3.3) dan persamaan (3.4) sebagai berikut :

$$F = \sum_{i=1}^{NG} F_i(P_i) \quad (3.3)$$

$$F_c(P_g) = \sum_{i=1}^{Ng} a_i + b_i P_{gi} + C_i P_{gi}^2 \quad (3.4)$$

Keterangan parameter persamaan (3.3) dan persamaan (3.4) sebagai berikut :

F : Nilai total biaya pembangkitan.

$F_i(P_i)$: Nilai biaya pembangkitan dari pembangkit ke – i yang merupakan fungsi dari output daya pembangkitan.

NG : Nilai jumlah dari unit pembangkit.

a_i, b_i, c_i : Nilai koefisien biaya pembangkit.

P_{gi} : Daya aktif yang dihasilkan generator.

3. 5. Penentuan *Constraint*

3. 5. 1. *Constraint Kapasitas Pembangkit*

Untuk kestabilan operasional, daya aktif masing – masing pembangkit dibatasi oleh batasan atas pembangkitan dan batasan bawah pembangkitan sebagai berikut :

$$P_{G1}^{min} \leq P_{G1} \leq P_{G1}^{max} \quad (3.5)$$

$$P_{G2}^{min} \leq P_{G2} \leq P_{G2}^{max} \quad (3.6)$$

$$P_{G3}^{min} \leq P_{G3} \leq P_{G3}^{max} \quad (3.7)$$

$$P_{G4}^{min} \leq P_{G4} \leq P_{G4}^{max} \quad (3.8)$$

$$P_{G5}^{min} \leq P_{G5} \leq P_{G5}^{max} \quad (3.9)$$

$$P_{G6}^{min} \leq P_{G6} \leq P_{G6}^{max} \quad (3.10)$$

Batasan pembangkitan daya reaktif pada masing – masing pembangkit sebagai berikut :

$$Q_{G1}^{min} \leq Q_{G1} \leq Q_{G1}^{max} \quad (3.11)$$

$$Q_{G2}^{min} \leq Q_{G2} \leq Q_{G2}^{max} \quad (3.12)$$

$$Q_{G3}^{min} \leq Q_{G3} \leq Q_{G3}^{max} \quad (3.13)$$

$$Q_{G4}^{min} \leq Q_{G4} \leq Q_{G4}^{max} \quad (3.14)$$

$$Q_{G5}^{min} \leq Q_{G5} \leq Q_{G5}^{max} \quad (3.15)$$

$$Q_{G6}^{min} \leq Q_{G6} \leq Q_{G6}^{max} \quad (3.16)$$

Keterangan parameter persamaan (3.6) sampai dengan persamaan (3.16) sebagai berikut:

- P_{Gi}^{min} : Daya aktif pembangkitan minimum masing – masing pembangkit (MW)
- P_{Gi}^{max} : Daya aktif pembangkitan maksimum masing – masing pembangkit (MW)
- Q_{Gi}^{min} : Daya reaktif pembangkitan minimum masing – masing pembangkit (MW)
- Q_{Gi}^{max} : Daya reaktif pembangkitan maksimum masing – masing pembangkit (MW)
- $Q_{shunt\ i}^{min}$: Batasan minimum dari *shunt capacitor* (MW)
- $Q_{shunt\ i}^{max}$: Batasan maksimum dari *shunt capacitor* (MW)

3. 5. 2. Keseimbangan Aliran Daya

Daya total pembangkitan harus sesuai dengan daya total beban dan losses pada jaringan transmisi. Persamaannya diberikan sebagai berikut :

$$\sum_{i=1}^n P_{gi} - P_D - P_{loss} \quad (3.17)$$

Keterangan parameter pada persamaan (3.17) :

P_D : Total beban (MW)

P_{loss} : Total Losses (MW)

3. 6. Penyelesaian *Multiobjective OPF* Menggunakan *Weighted Sum Method*

Weighted sum method merupakan metode yang menggabungkan beberapa fungsi obyektif menjadi satu kesatuan, dengan cara perkalian masing – masing fungsi obyektif dengan koefisien *weight*. Nilai *weight* dari masing – masing *objective function* merupakan nilai yang ditentukan dari proporsi relatif dari masing – masing *objective function*.

Bentuk umum kombinasi dari fungsi multiobyektif yang digunakan sebagai berikut :

$$F = w_1 f_1 + w_2 f_2 \quad (3.18)$$

$$w_1 + w_2 = 1 \quad (3.19)$$

Penalty factor untuk masing – masing fungsi obyektif dipertimbangkan didalam persamaan sebagai berikut :

$$F = w_1 \alpha f_1 + w_2 \beta f_2 \quad (3.20)$$

$$\alpha = \frac{f_{ref\ max}}{f_1\ max} \quad (3.21)$$

$$\beta = \frac{f_{ref\ max}}{f_2\ max} \quad (3.22)$$

Dengan mensubstitusi persamaan (3.18), persamaan (3.20), persamaan (3.21), dan persamaan (3.22) maka akan didapatkan nilai masing – masing *penalty factor* sebagai berikut :

$$\alpha = \frac{f_{ref\ max}}{\sum_{k=1}^{Nl} P_{l_1\ max}} \quad (3.23)$$

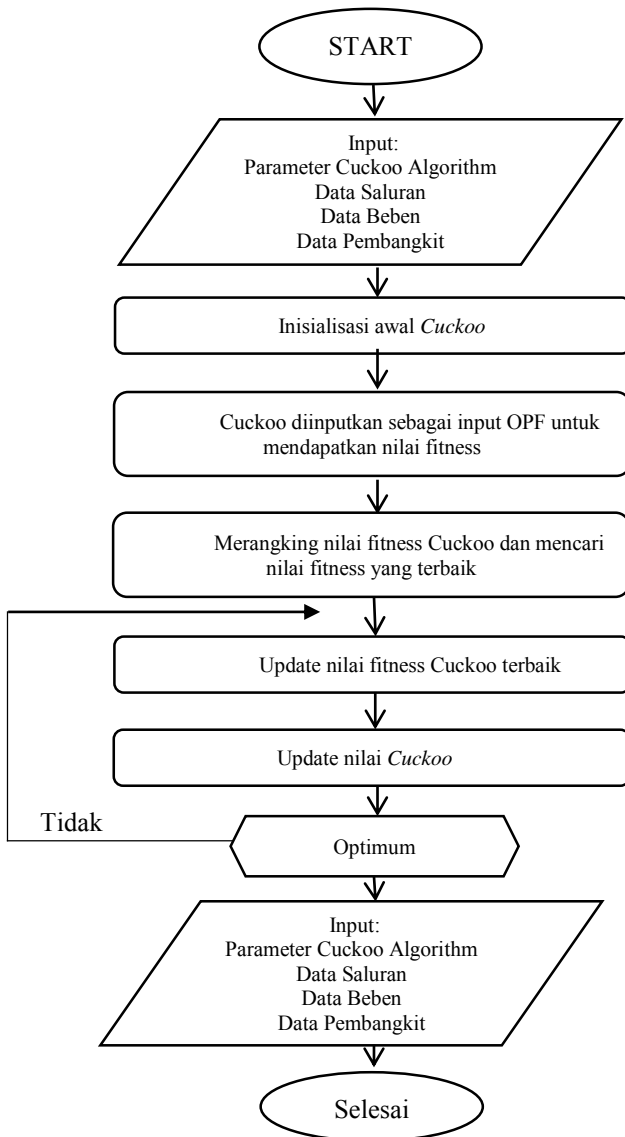
$$\beta = \frac{f_{ref\ max}}{\sum_{i=1}^{Ng} a_i + b_i P_{gi} + c_i P_{gi}^2} \quad (3.24)$$

Keterangan parameter persamaan (3.20) sampai dengan persamaan (3.24) sebagai berikut:

- F : *Multiobjective function*
- w_1 : *Weight objective function 1*
- α : *Factor normalisasi objective function 1*
- f_1 : *Objective function 1*
- w_2 : *Weight objective function 2*
- β : *Factor normalisasi objective function 2*
- f_2 : *Objective function 2*

3. 7. Simulasi Algoritma Cuckoo/ *Cuckoo Search Algorithm* untuk Menyelesaikan *Multiobjective OPF*

Dalam tugas akhir ini untuk mendapatkan kombinasi dari beberapa nilai parameter yang optimal digunakan algoritma *Cuckoo*. pencarian hasil yang unik tetapi memiliki kecepatan dan ketepatan hasil yang tinggi. *Cuckoo Search Algorithm* memuat n(banyak), masing-masing *Cuckoo* memuat nilai variable-variable yang akan merandom Diagram alir penyelesaian OPF dengan menggunakan *Cuckoo search Alghorithm* diberikan pada gambar 3.2



Gambar 3.2 : Flowchart penyelesaian OPF menggunakan Cuckoo Algorithm

3. 7. 1. Inisialisasi Cuckoo

Pada tahap ini, populasi dari cuckoo diambil secara acak diantara batas atas dan batas bawah yang sudah ditentukan. Pada permasalahan OPF yang kompleks, populasi cuckoo yang diambil mewakili tegangan (V) setiap pembangkit dan daya aktif (P) setiap pembangkit. Sedangkan pada OPF yang sederhana, cuckoo mewakili *variable* daya aktif (P).

Proses inisialisasi pembatasan Cuckoo sesuai dengan *constraint* batas bawah dan batas atas yang diberikan sebagai berikut :

$$P^{k_{ij}} = \begin{cases} P^{k_{ij}} & \text{if } P_{ij,\min} \leq P^{k_{ij}} \leq P_{ij,\max} \\ P^{k_{ij},\min} & \text{if } P^{k_{ij}} < P_{ij,\min} \\ P^{k_{ij},\max} & \text{if } P^{k_{ij}} > P_{ij,\max} \end{cases}$$

Dengan langkah ini didapatkan cuckoo yang memiliki range yang telah memenuhi batas atas dan batas bawah. jika ada individu Cuckoo yang memiliki titik operasi diatas titik operasi maksimum, maka akan diset titik maksimum yang diperbolehkan. Jika terdapat individu Cuckoo yang memiliki titik operasi dibawah titik operasi minimum, maka akan diset titik minimum yang diperbolehkan.

3. 7. 2. Pergerakan Cuckoo with Levy Flight

Pergerakan Cuckoo mencari dari host i ke host j yang memiliki kualitas inang yang lebih baik diberikan persamaan (3.25)

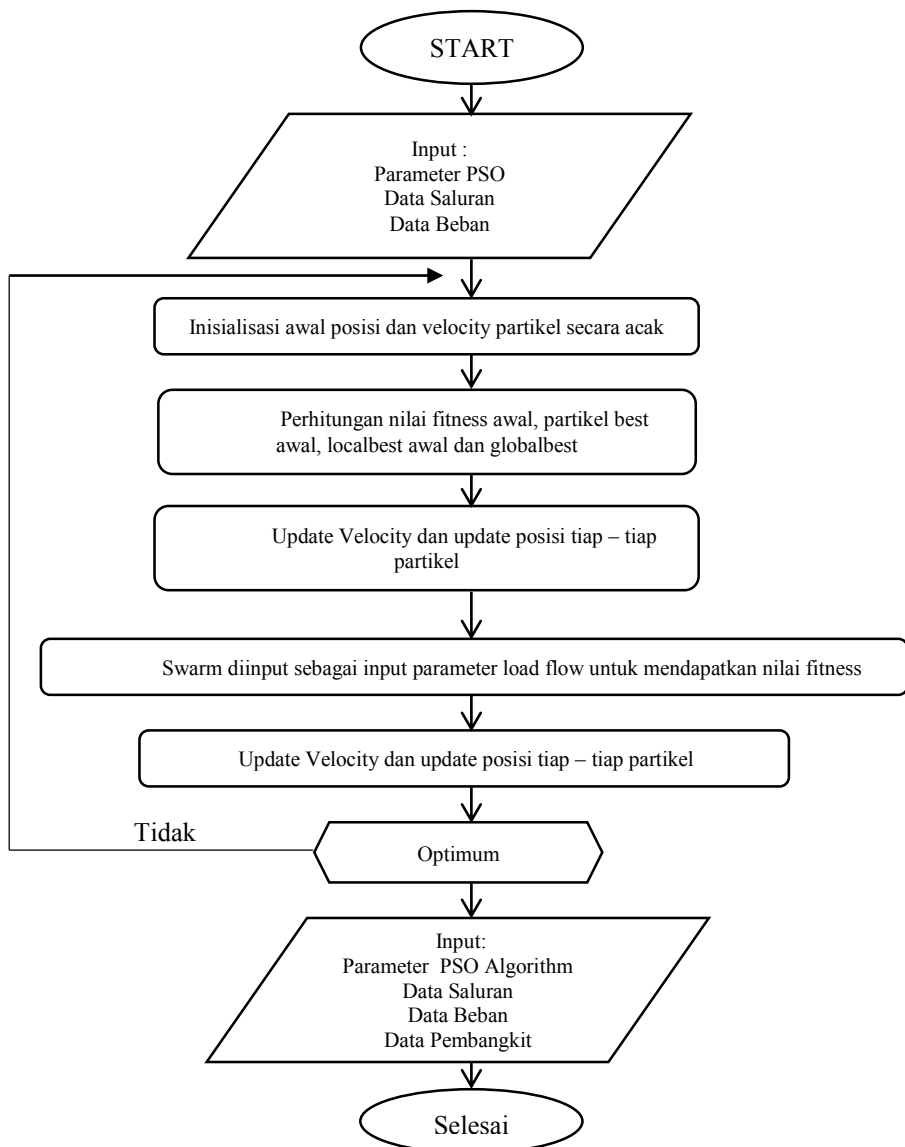
$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus L'evy(\lambda) \quad (3.25)$$

untuk mengatur *random walk* dari populasi Cuckoo diberikan persamaan (3.26)

$$L'evy \sim u = t - \lambda, (1 < \lambda \leq 3) \quad (3.26)$$

3. 8. Simulasi Particle Swarm Optimization (PSO) untuk Menyelesaikan Multiobjective OPF

Tugas akhir ini digunakan simulasi *Particle Swarm Optimization* (PSO) sebagai metode sekunder dalam menyelesaikan permasalahan *multiobjective* OPF . Diagram alur penyelesaian OPF dengan menggunakan PSO diberikan pada gambar 3.3.



Gambar 3.2 : Flowchart penyelesaian OPF menggunakan PSO

3.8.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization adalah salah satu metode optimasi yang terinspirasi dari perilaku gerakan kawanan hewan seperti ikan (*school of fish*), hewan *herbivor* (*herd*), dan burung (*flock*) yang kemudian tiap objek hewan disederhanakan menjadi sebuah partikel. Suatu partikel dalam ruang memiliki posisi yang dikodekan sebagai vektor koordinat. Vektor posisi ini dianggap sebagai keadaan yang sedang ditempati oleh suatu partikel di ruang pencarian. Setiap posisi dalam ruang pencarian merupakan alternatif solusi yang dapat dievaluasi menggunakan fungsi objektif. Setiap partikel bergerak dengan kecepatan v .

Particle Swarm Optimization atau yang kita kenal dengan PSO menerapkan sifat masing-masing individu dalam satu kelompok besar. Kemudian menggabungkan sifat-sifat tersebut untuk menyelesaikan permasalahan. Particle Swarm Optimization pertama kali dimunculkan pada tahun 1995, sejak saat itulah para peneliti banyak menurunkan dan mengembangkan metode PSO.

Ciri khas dari PSO adalah pengaturan kecepatan partikel secara heuristik dan probabilistik. Jika suatu partikel memiliki kecepatan yang konstan maka jika jejak posisi suatu partikel divisualisasikan akan membentuk garis lurus. Dengan adanya faktor eksternal yang membelokkan garis tersebut yang kemudian menggerakkan partikel dalam ruang pencarian maka diharapkan partikel dapat mengarah, mendekati, dan pada akhirnya mencapai titik optimal. Faktor eksternal yang dimaksud antara lain posisi terbaik yang pernah dikunjungi suatu partikel, posisi terbaik seluruh partikel (diasumsikan setiap partikel mengetahui posisi terbaik setiap partikel lainnya), serta faktor kreativitas untuk melakukan eksplorasi.

Particle Swarm Optimization memiliki kesamaan sifat dengan teknik komputasi seperti Algoritma Genetika (Genetic Algorithm). Sistem PSO diinisialisasi oleh sebuah populasi solusi secara acak dan selanjutnya mencari titik optimum dengan cara meng-update tiap hasil pembangkitan. Metode optimasi yang didasarkan pada swarm intelligence ini disebut algoritma behaviorally inspired sebagai alternatif dari algoritma genetika, yang sering disebut evolution-based procedures. Dalam konteks optimasi multivariabel, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Setiap partikel diasumsikan memiliki dua karakteristik: posisi dan

kecepatan. Setiap partikel bergerak dalam ruang/space tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut.

Sebagai contoh, misalnya perilaku burung-burung dalam kawanan burung. Meskipun setiap burung mempunyai keterbatasan dalam hal kecerdasan, biasanya ia akan mengikuti kebiasaan (rule) seperti berikut :

1. Seekor burung tidak berada terlalu dekat dengan burung yang lain
2. Burung tersebut akan mengarahkan terbangnya ke arah rata-rata keseluruhan burung
3. Akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor simpel berikut:

1. Kohesi - terbang bersama
2. Separasi - jangan terlalu dekat
3. Penyesuaian (*alignment*) - mengikuti arah bersama

Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu :

$$X_{it}(t) = x_{i1}(t), x_{i2}(t), \dots, x_{iN}(t) \quad (3.27)$$

$$V_i(t) = v_{i1}(t), v_{i2}(t), \dots, v_{iN}(t) \quad (3.28)$$

keterangan persamaan (3.27) dan (3.28) sebagai berikut :

X= posisi partikel

V= kecepatan partikel

i= indeks partikel

t= iterasi ke-t

N= ukuran dimensi ruang

Berikut ini merupakan model matematika yang menggambarkan mekanisme updating status partikel Kennedy and Eberhart [1995]:

$$V_i(t) = V_i(t - 1) + c1r1(X_i^L - X_i(t - 1)) + c2r2 (X^G - X_i(t - 1))$$

$$X_i(t) = V_i(t) + X_i(t - 1)$$

Dimana

$X_i^L = X_{i1}^L, X_{i2}^L, \dots, X_{iN}^L$ Merepresentasikan local best dari partikel ke-i. Sedangkan, $X_i^G = X_1^G, X_2^G, \dots, X_N^G$ merepresentasikan global best dari seluruh kawanan. Sedangkan c1 dan c2 adalah suatu konstanta yang bernilai positif yang biasanya disebut sebagai *learning factor*. Kemudian r1 dan r2 adalah suatu bilangan *random* yang bernilai antara 0 sampai 1, digunakan untuk menghitung kecepatan partikel yang baru berdasarkan kecepatan sebelumnya, jarak antara posisi saat ini dengan posisi terbaik partikel (*local best*), dan jarak antara posisi saat ini dengan posisi terbaik kawanan (*global best*). Kemudian partikel terbang menuju posisi yang baru berdasarkan persamaan(2.). Setelah algoritma PSO ini dijalankan dengan sejumlah iterasi tertentu hingga mencapai kriteria pemberhentian, maka akan didapatkan solusi yang terletak pada *global best*. Model ini akan disimulasikan dalam ruang dengan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi). Ini dilakukan hingga maksimum iterasi dicapai atau bisa juga digunakan kriteria penghentian yang lain.

BAB 4

SIMULASI DAN ANALISIS

Pada skripsi ini dilakukan simulasi OPF dengan berbagai permasalahan untuk menentukan minimisasi biaya dan emisi dari pembangkit. Metode utama yang digunakan untuk menyelesaikan permasalahan *multiobjective* OPF adalah algoritma Cuckoo (*Cuckoo Search Algorithm*). Metode lain yang digunakan sebagai pembanding untuk menyelesaikan permasalahan OPF pada skripsi ini adalah *Particle Swarm Optimization* (PSO).

Simulasi yang dikerjakan pada skripsi ini adalah :

1. Simulasi OPF menggunakan satu
 - a. Simulasi *Economic Dispatch*
 - b. Simulasi *Emission Minimization*
2. Simulasi OPF menggunakan dua fungsi obyektif
 - a. Simulasi *Economic Emission Minimization*

4. 1. Data IEEE 30-Bus

Dalam skripsi ini digunakan sistem standar IEEE 30 bus yang terdiri dari 6 unit pembangkit yang berada pada bus 1, bus 2, bus 5, bus 8, bus 11, bus 13. Tabel 4.1 menyajikan data sistem standar IEEE 30 bus. Tabel 4.1 menyajikan data saluran IEEE 30 bus.

Tabel 4.1 : Data IEEE 30 Bus

Bus	Bus	Voltage	Angle	Load		Generator				Static Mvar
No	Code	Mag.	Degree	MW	Mvar	MW	Mvar	Qmin	Qmax	+Qc / - Q1
1	1	1,06	0	0	0	0	0	0	0	0
2	2	1,043	0	2,17	12,7	40	0	-40	50	0
3	0	1	0	2,4	1,2	0	0	0	0	0
4	0	1,06	0	7,6	1,6	0	0	0	0	0
5	2	1,01	0	94,2	19	0	0	-40	40	0
6	0	1	0	0	0	0	0	0	0	0
7	0	1	0	22,8	10,9	0	0	0	0	0
8	2	1,01	0	30	30	0	0	-30	40	0
9	0	1	0	0	0	0	0	0	0	0
10	0	1	0	5,8	2	0	0	-6	24	0
11	2	1,082	0	0	0	0	0	0	0	0

Lanjutan tabel 4.1 : Data IEEE 30 bus

Bus	Bus	Voltage	Angle	Load		Generator				Static Mvar
No	Code	Mag.	Degree	MW	Mvar	MW	Mvar	Qmin	Qmax	+Qc / -Q1
14	0	1	0	6,2	1,6	0	0	0	0	0
15	0	1	0	8,2	2,5	0	0	0	0	0
16	0	1	0	3,5	1,8	0	0	0	0	0
17	0	1	0	9	5,8	0	0	0	0	0
18	0	1	0	3,2	0,9	0	0	0	0	0
19	0	1	0	9,5	3,4	0	0	0	0	0
20	0	1	0	2,2	0,7	0	0	0	0	0
21	0	1	0	17,5	11,2	0	0	0	0	0
22	0	1	0	0	0	0	0	0	0	0
23	0	1	0	3,2	1,6	0	0	0	0	0
24	0	1	0	8,7	6,7	0	0	0	0	0
25	0	1	0	0	0	0	0	0	0	0
26	0	1	0	3,5	2,3	0	0	0	0	0
27	1	0	0	0	0	0	0	0	0	0
28	1	0	0	0	0	0	0	0	0	0
29	1	0	0	2,4	0,9	0	0	0	0	0
30	1	0	0	10,6	1,9	0	0	0	0	0

Tabel 4.2 : Data saluran IEEE 30 Bus

Bus	Bus	R	X	$\frac{1}{2} b$
nl	nr	Pu	Pu	pu
1	2	0,0192	0,0575	0,0264
1	3	0,0452	0,1852	0,0204
2	4	0,057	0,1737	0,0184
3	4	0,0132	0,0379	0,0042
2	5	0,0472	0,1983	0,0209
2	6	0,0581	0,1763	0,0187
4	6	0,0119	0,0414	0,0045
5	7	0,046	0,116	0,0102
6	7	0,0267	0,082	0,0085
6	8	0,012	0,042	0,0045
6	9	0	0,208	0,97
6	10	0	0,556	0,96
9	11	0	0,208	0
9	10	0	0,11	0
4	12	0	0,256	0,93
12	13	0	0,14	0
12	14	0,1231	0,2559	0
12	15	0,0662	0,1304	0

Lanjutan tabel 4.2 : Data Saluran IEEE 30 Bus

Bus	Bus	R	X	$\frac{1}{2} b$
nl	nr	Pu	pu	Pu
12	16	0,0945	0,1987	0
14	15	0,221	0,1997	0
16	17	0,0824	0,1923	0
15	18	0,1073	0,2185	0
18	19	0,0693	0,1292	0
19	20	0,034	0,068	0
10	20	0,0936	0,209	0
10	17	0,0324	0,0845	0
10	21	0,0348	0,0749	0
10	22	0,0727	0,1499	0
21	22	0,0116	0,0236	0
15	23	0,1	0,202	0
22	24	0,115	0,179	0
23	24	0,132	0,27	0
24	25	0,1885	0,3292	0
25	26	0,2544	0,38	0
25	27	0,1093	0,2087	0
28	27	0	0,396	0,96
27	29	0,2198	0,4153	0
27	30	0,3202	0,6027	0
29	30	0,2399	0,4533	0
8	28	0,0636	0,2	0,0214
6	28	0,0169	0,0599	0,065

Tabel 4.3. Data Generator IEEE 30 Bus

No	Pembangkit	Daya Aktif (MW)		Daya Reaktif (MVAR)	
		Minimum	Maksimum	Minimum	Maksimum
1	Pembangkit 1	50	200	-20	200
2	Pembangkit 2	20	80	-20	100
3	Pembangkit 3	15	50	-15	80
4	Pembangkit 4	10	35	-10	60
5	Pembangkit 5	10	30	-10	50
6	Pembangkit 6	12	40	-10	60

Persamaan karakteristik biaya pembangkitan pada masing – masing pembangkit sistem IEEE 30 Bus ditunjukkan pada tabel 4.4 berikut.

Tabel 4.4. Data Spesifikasi CPU

No	Pembangkit	Fungsi Emisi (kg / Jam)
1	Pembangkit 1 (Bus 1)	$E_1 = 0.0126 P_1^2 + 0.1 P_1 + 22.983$
2	Pembangkit 2 (Bus 2)	$E_2 = 0.02P_2^2 + 0.1 P_2 + 22.213$
3	Pembangkit 3 (Bus 5)	$E_3 = 0.0270P_3^2 + (-0.1) P_3 + 25.505$
4	Pembangkit 4 (Bus 8)	$E_4 = 0.0291P_4^2 + (-0.005)P_4 + 24.900$
5	Pembangkit 5 (Bus 11)	$E_5 = 0.0290P_5^2 + (-0.4)P_5 + 24.700$
6	Pembangkit 6 (Bus 13)	$E_6 = 0.0271 P_6^2 + (-0.0055)P_6 + 25.300$

Dalam melakukan simulasi, komputer yang digunakan memiliki spesifikasi yang ditunjukkan pada tabel 4.7 sebagai berikut :

Tabel 4.5. Data Spesifikasi CPU

No	Parameter	Spesifikasi CPU
1	Processor	Core i7
2	Speed Processing	3,4 ghz
3	Installed Memory (RAM)	8 GB
4	System Type	Windows 8

4. 2. Simulasi *Economic Dispatch*

Economic Dispatch (ED) akan disimulasikan menggunakan satu program utama yaitu algoritma Cuckoo dan *Particle Swarm Optimization* (PSO)

4. 2. 1. Simulasi *Economic Dispatch Cuckoo*

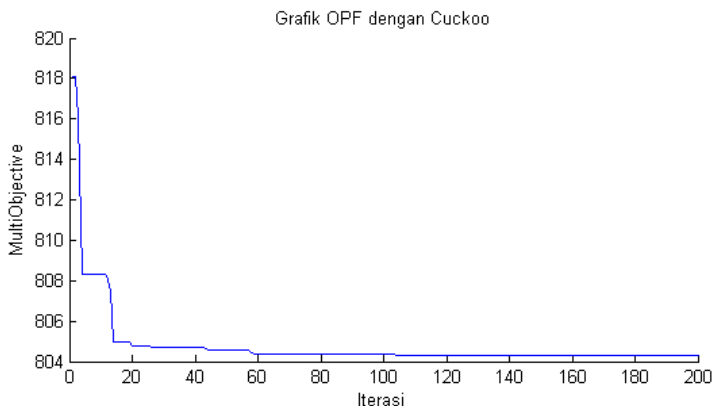
Parameter yang digunakan untuk mengimplementasikan Cuckoo search Algorithm dalam menyelesaikan OPF *Economic Dispatch* pada sistem IEEE 30 bus ditunjukkan pada tabel 4.9

Tabel 4.6. Parameter Cuckoo

Jumlah Partikel	Jumlah Iterasi
25	200

Parameter – parameter pada tabel 4.9 digunakan secara konsisten dalam keseluruhan simulasi *Economic Dispatch* cuckoo. Jumlah partikel menyatakan jumlah penyimpan solusi. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi cuckoo untuk penyelesaian *Economic Dispatch* ditunjukkan pada gambar 4.1.



Gambar 4.1. Grafik Konvergensi Economic Dispatch menggunakan metode Cuckoo

Daya aktif dan biaya pembangkitan hasil simulasi *Economic Dispatch* sistem IEEE 30 bus menggunakan metode Cuckoo ditunjukan tabel 4.10. Pada simulasi *Economic Dispatch*, Cuckoo Konvergen pada iterasi ke 105 mencapai titik konvergen dengan nilai fitness 802,9875, total daya aktif pembangkitan sebesar 293,006 MW, total biaya pembangkitan sebesar 803,02 \$/jam dan total emisi sebesar 592,71 Kg/jam.

Tabel 4.7. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi *Economic Dispatch* menggunakan metode cuckoo

No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	175,695	467,148	394,360
2	Daya Pembangkit 2	48,914	127,469	65,273
3	Daya Pembangkit 3	21,857	51,716	36,218
4	Daya Pembangkit 4	22,283	76,559	39,237
5	Daya Pembangkit 5	12,257	40,527	28,567
6	Daya Pembangkit 6	12,000	39,600	29,136
	Total			

4. 2. 2. Simulasi *Economic Dispatch* PSO

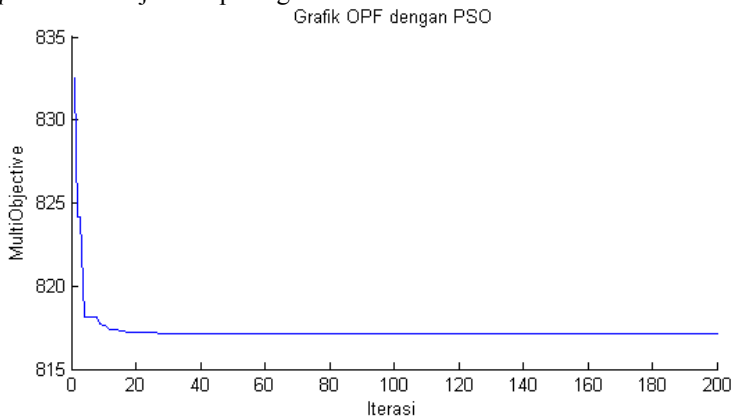
Parameter yang digunakan untuk mengimplementasikan PSO dalam menyelesaikan OPF *Economic Dispatch* pada sistem IEEE 30 bus ditunjukkan pada tabel 4.11

Tabel 4.8. Parameter PSO

Jumlah Partikel	Jumlah Iterasi
25	200

Parameter – parameter pada tabel 4.9 digunakan secara konsisten dalam keseluruhan simulasi *Economic Dispatch* PSO. Jumlah partikel menyatakan jumlah penyimpanan solusi OPF. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi PSO untuk penyelesaian OPF *Economic Dispatch* ditunjukkan pada gambar 4.2



Gambar 4.2. Grafik Konvergensi *Economic Dispatch* menggunakan metode PSO

Daya aktif dan biaya pembangkitan hasil simulasi *Economic Dispatch* sistem IEEE 30 bus menggunakan metode PSO ditunjukkan tabel 4.10. Pada simulasi ini, PSO konvergen pada iterasi ke 105 mencapai titik konvergen dengan nilai fitness 810,3685, total daya aktif pembangkitan sebesar 293.469 MW, total biaya pembangkitan sebesar 810,35 \$/jam dan total emisi sebesar 559,359 Kg/jam.

Tabel 4.9. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi Economic Dispatch menggunakan metode cuckoo

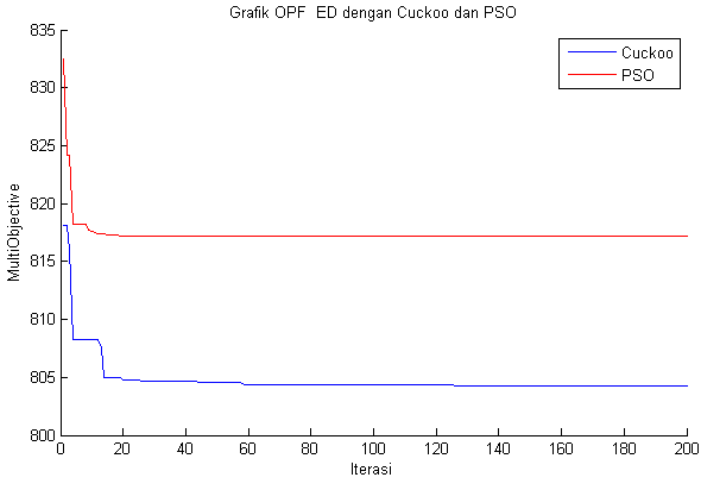
No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	166,799	437,931	356,860
2	Daya Pembangkit 2	44,908	113,883	58,158
3	Daya Pembangkit 3	21,944	52,041	36,313
4	Daya Pembangkit 4	21,886	75,126	38,730
5	Daya Pembangkit 5	20,001	70,003	35,501
6	Daya Pembangkit 6	17,881	61,362	33,799
	Total			

4. 2. 3. Perbandingan Hasil *Economic Dispatch* Menggunakan Cuckoo dan PSO

Perbandingan hasil simulasi *Economic Dispatch* menggunakan beberapa metode optimisasi : Cuckoo dan PSO diberikan pada tabel 4.13.

Tabel 4.10. Perbandingan Hasil Simulasi Economic Dispatch Menggunakan Metode Cuckoo dan PSO

Metode	Nilai Fitness Terbaik	Biaya Total Pembangkitan (\$/Jam)
Cuckoo	802,9875	803,02
PSO	810,3685	810,35



Gambar 4.3. Perbandingan Nilai Konvergensi Cuckoo dan PSO pada simulasi Economic Dispatch

Dari penyelesaian *Economic Dispatch* pada sistem test IEEE 30 bus, metode Cuckoo menghasilkan nilai perhitungan yang paling ekonomis, yaitu 803,02 \$/jam. Perbandingan konvergensi dari metode Cuckoo dan PSO ditunjukkan pada gambar 4.3. PSO mencapai titik konvergen pada iterasi ke 30 dengan nilai fitness 802,9875. Cuckoo mencapai titik konvergen pada iterasi ke 60 dengan nilai fitness 810,3685

4. 3. Simulasi *Emission Minimization*

Emission Minimization akan disimulasikan mennggunakan satu program utama yaitu algoritma Cuckoo dan *Particle Swarm Optimation* (PSO)

4. 3. 1. Simulasi OPF *Economic Emission Minimization Cuckoo*

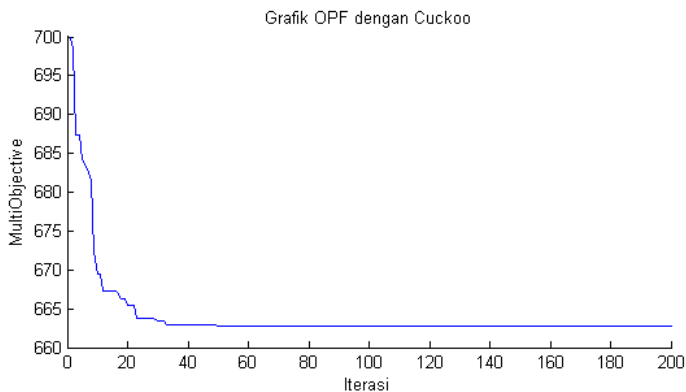
Parameter yang digunakan untuk mengimplementasikan Cuckoo search Algorithm dalam menyelesaikan *Emission Minimization* pada sistem IEEE 30 bus ditunjukkan pada tabel 4.14

Tabel 4.11. Parameter Cuckoo

Jumlah Partikel	Jumlah Iterasi
25	200

Parameter – parameter pada tabel 4.14 digunakan secara konsisten dalam keseluruhan simulasi *Emission Minimization* cuckoo. Jumlah partikel menyatakan jumlah penyimpanan solusi OPF. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi cuckoo untuk penyelesaian *Emission Minimization* ditunjukkan pada gambar 4.4.



Gambar 4.4. Grafik Konvergensi *Emission Minimization* menggunakan metode Cuckoo

Daya aktif dan biaya pembangkitan hasil simulasi *Emission Minimizati* sistem IEEE 30 bus menggunakan metode Cuckoo ditunjukkan tabel 4.15. Pada simulasi OPF *Emission Minimization*, Cuckoo konvergen pada iterasi ke 50 mencapai titik konvergen dengan nilai fitness 662,7418, total daya aktif pembangkitan sebesar 287,893 MW, total biaya pembangkitan sebesar 897,51 \$/jam dan total emisi sebesar 432,011 Kg/jam.

Tabel 4.12. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi Emission Minimization menggunakan metode cuckoo

No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	85,907	199,488	107,380
2	Daya Pembangkit 2	55,240	150,072	77,819
3	Daya Pembangkit 3	42,392	154,707	69,786
4	Daya Pembangkit 4	35,000	123,967	60,373
5	Daya Pembangkit 5	30,000	112,500	49,600
6	Daya Pembangkit 6	39,354	156,778	67,057
	Total			

4. 3. 2. Simulasi OPF *Economic Emission Minimization* PSO

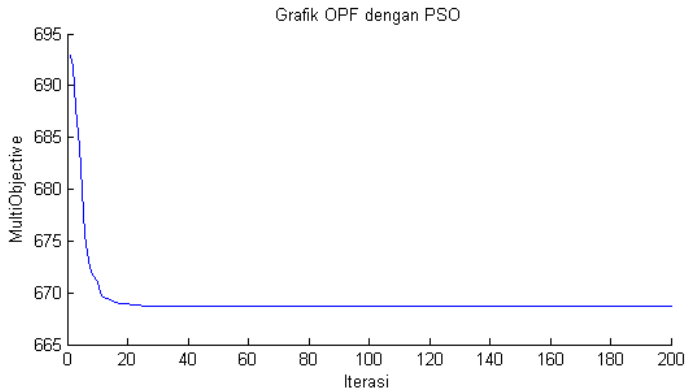
Parameter yang digunakan untuk mengimplementasikan PSO search Algorithm dalam menyelesaikan OPF *Emission Minimization* pada sistem IEEE 30 bus ditunjukkan pada tabel 4.16

Tabel 4.13. Parameter PSO

Jumlah Partikel	Jumlah Iterasi
25	200

Parameter – parameter pada tabel 4.14 digunakan secara konsisten dalam keseluruhan simulasi OPF *Emission Minimization* PSO. Jumlah partikel menyatakan jumlah penyimpanan solusi OPF. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi PSO untuk penyelesaian OPF *Emission Minimization* ditunjukkan pada gambar 4.4.



Gambar 4.5. Grafik Konvergensi OPF Emission Minimization menggunakan metode PSO

Daya aktif dan biaya pembangkitan hasil simulasi OPF *Emission Minimizati* sistem IEEE 30 bus menggunakan metode PSO ditunjukan tabel 4.15. Pada simulasi OPF *Emission Minimization*, Cuckoo konvergen pada iterasi ke 30 mencapai titik konvergen dengan nilai fitness 668,7204, total daya aktif pembangkitan sebesar 287,979 MW, total biaya pembangkitan sebesar 884,48 \$/jam dan total emisi sebesar 435,924 Kg/jam.

Tabel 4.14. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF Emission Minimization menggunakan metode PSO

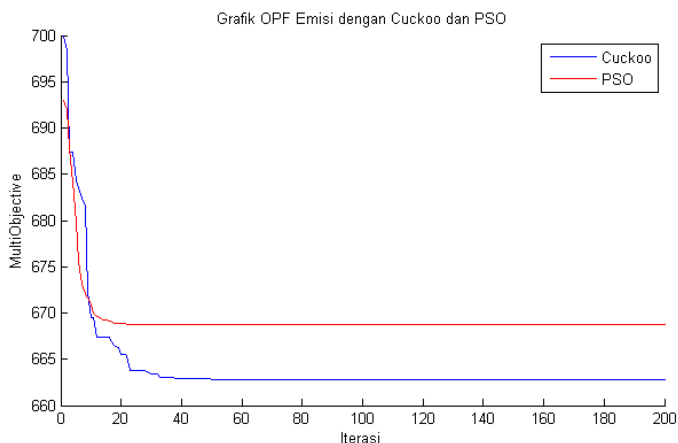
No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	89,524	209,103	115,014
2	Daya Pembangkit 2	64,261	184,723	98,447
3	Daya Pembangkit 3	38,616	131,816	61,906
4	Daya Pembangkit 4	26,779	93,012	45,634
5	Daya Pembangkit 5	32,117	122,141	53,330
6	Daya Pembangkit 6	36,682	143,585	61,563
	Total			

4. 3. 3. Perbandingan Hasil OPF *Emission Minimization* Menggunakan Cuckoo dan PSO

Perbandingan hasil simulasi OPF *Emission Minimization* menggunakan beberapa metode optimisasi : Cuckoo dan PSO diberikan pada tabel 4.18.

Tabel 4.15. Perbandingan Hasil Simulasi OPF *Emission Minimization* Menggunakan Metode Cuckoo dan PSO

Metode	Nilai Fitness Terbaik	Emisi (Kg/jam)
Cuckoo	662,7418	432,011
PSO	668,7204	435,924



Gambar 4.6. Perbandingan Nilai Konvergensi Cuckoo dan PSO pada simulasi OPF *Emission Minimization*

Dari penyelesaian OPF *Emission Minimization* pada sistem tes IEEE 30 bus, metode Cuckoo menghasilkan nilai perhitungan yang paling minimal, yaitu 432,011 (Kg/jam). Perbandingan konvergensi dari metode Cuckoo dan PSO ditunjukkan pada gambar 4.6. PSO mencapai titik konvergen pada iterasi ke 30 dengan nilai fitness 668,7204. Cuckoo mencapai titik konvergen pada iterasi ke 60 dengan nilai fitness 662,7418

4. 4. Simulasi *Optimal Power Flow Economic Emission Dispatch*

Optimal power flow akan disimulasikan menggunakan satu program utama yaitu algoritma Cuckoo dan *Particle Swarm Optimization* (PSO)

4. 4. 1. Simulasi *OPF Economic Emission Dispatch using Cuckoo*

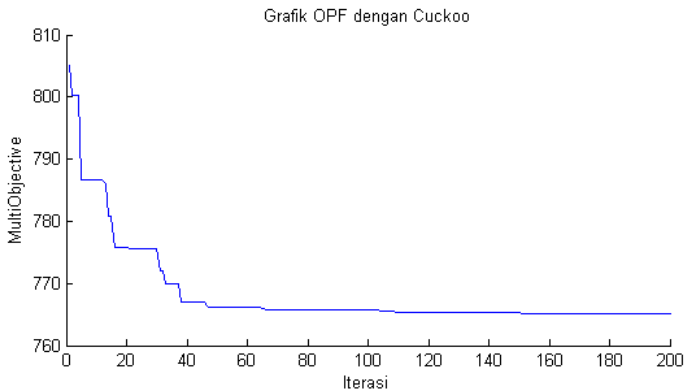
Parameter yang digunakan untuk mengimplementasikan algoritma Cuckoo dalam menyelesaikan OPF pada sistem IEEE 30 bus ditunjukkan pada tabel 4.19

Tabel 4.16. Parameter Cuckoo

Jumlah Partikel	Jumlah Iterasi
25	500

Parameter – parameter pada tabel 4.19 digunakan secara konsisten dalam keseluruhan simulasi OPF cuckoo. Jumlah partikel menyatakan jumlah penyimpanan solusi OPF. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi cuckoo untuk penyelesaian OPF ditunjukkan pada gambar 4.7.



Gambar 4.7. Grafik konvergensi simulasi *economic emission dispatch* menggunakan metode Cuckoo

Daya aktif, biaya pembangkitan dan emisi hasil simulasi OPF sistem IEEE 30 bus menggunakan metode Cuckoo ditunjukkan tabel 4.20. Pada simulasi OPF, Cuckoo konvergen pada iterasi ke 150 mencapai titik konvergen dengan nilai fitness 765,0801, total daya aktif

pembangkitan sebesar 288,916 MW, total biaya pembangkitan sebesar 850,99 \$/jam dan total emisi sebesar 442,867 Kg/jam.

Tabel 4.17. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF menggunakan metode cuckoo

No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	106,458	255,415	155,136
2	Daya Pembangkit 2	55,414	150,712	78,186
3	Daya Pembangkit 3	31,300	92,530	48,827
4	Daya Pembangkit 4	35,000	123,967	60,373
5	Daya Pembangkit 5	30,000	112,500	49,600
6	Daya Pembangkit 6	30,744	115,863	50,746

4. 4. 2. Simulasi OPF *Economic Emission Dispatch using PSO*

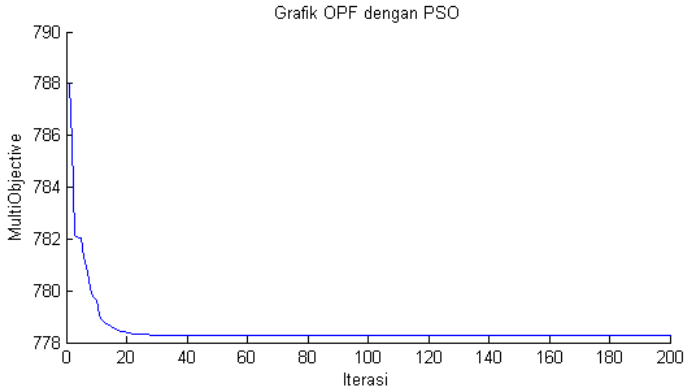
Parameter yang digunakan untuk mengimplementasikan PSO dalam menyelesaikan OPF pada sistem IEEE 30 bus ditunjukkan pada tabel 4.21

Tabel 4.18. Parameter PSO

Jumlah Partikel	Jumlah Iterasi
25	500

Parameter – parameter pada tabel 4.21 digunakan secara konsisten dalam keseluruhan simulasi OPF PSO. Jumlah partikel menyatakan jumlah penyimpanan solusi OPF. Jumlah iterasi merupakan jumlah total evaluasi fungsi secara keseluruhan.

Grafik Konvergensi PSO untuk penyelesaian OPF ditunjukkan pada gambar 4.8



Gambar 4.8. Grafik konvergensi economic emission dispatch menggunakan PSO

Daya aktif dan biaya pembangkitan hasil simulasi OPF sistem IEEE 30 bus menggunakan metode PSO ditunjukkan tabel 4.12. Pada simulasi OPF, PSO konvergen pada iterasi ke 35 mencapai titik konvergen dengan nilai fitness 778,2874, total daya aktif pembangkitan sebesar 289,938 MW, total biaya pembangkitan sebesar 853,39 \$/jam dan total emisi sebesar 470,101 Kg/jam.

Tabel 4.19. Daya aktif, Biaya Pembangkitan dan Emisi hasil simulasi OPF menggunakan metode cuckoo

No	Pembangkitan	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
1	Daya Pembangkit 1	121,920	299,583	198,085
2	Daya Pembangkit 2	63,153	180,315	95,765
3	Daya Pembangkit 3	34,225	107,432	53,708
4	Daya Pembangkit 4	28,607	99,798	48,571
5	Daya Pembangkit 5	22,289	79,287	38,216
6	Daya Pembangkit 6	19,744	68,977	35,756

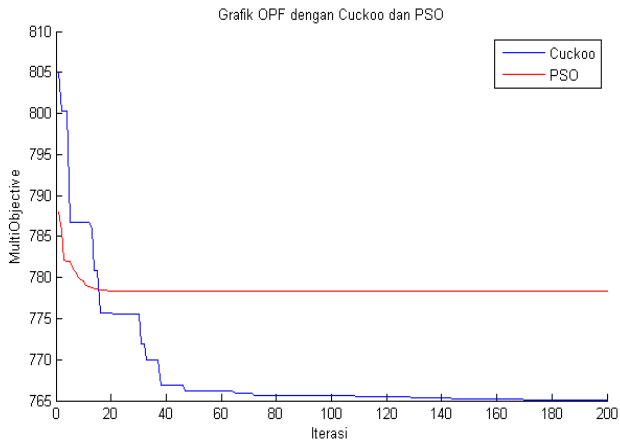
4. 4. 3. Perbandingan Hasil Simulasi OPF Economic Emission Dispatch using Cuckoo and PSO

Perbandingan Hasil Simulasi OPF Economic Emission Dispatch Menggunakan Metode heuristic : Cuckoo dan PSO pada tabel 4.20

Tabel 4.20. Perbandingan Hasil Simulasi OPF Economic Emission Dispatch Menggunakan metode Cuckoo Search dan PSO

Metode	Nilai Konvergensi Terbaik	Total daya Pembangkitan (MW)	Biaya (\$/jam)	Emisi (Kg/jam)
Cuckoo	765,0801	288,916	850,99	442,867
PSO	778,2874	289,938	853,39	470,101

Perbandingan nilai konvergensi metode Cuckoo Search dan PSO pada simulasi OPF *Economic Emission Dispatch* ditunjukkan pada gambar 4.9. Pada iterasi ke 70, Cuckoo memiliki fitness yang lebih baik dari nilai fitness metode PSO yaitu :765,0801. PSO konvergen pada iterasi ke 35 dengan nilai fitness 778,2874. Dari kedua metode diatas penyelesaian *Economic Emission Dispatch* pada system IEEE 30 bus, fitness terbaik dihasilkan oleh metode Cuckoo yaitu sebesar 765,0801.



Gambar 4.9 : Perbandingan grafik konvergensi economic emmision dispatch menggunakan metode Cuckoo dan PSO

BAB 5

KESIMPULAN DAN SARAN

5. 1. Kesimpulan

1. Dalam simulasi OPF dengan satu fungsi obyektif, Cuckoo memberikan nilai fitness yang paling baik dibandingkan dengan metode PSO. Nilai fitness Cuckoo pada simulasi *Economic dispatch* mencapai 765,0801
2. Dalam simulasi OPF dengan satu fungsi obyektif, Cuckoo memberikan nilai fitness yang paling baik dibandingkan dengan metode PSO. Nilai fitness Cuckoo pada simulasi *Emission Minimization* mencapai 778,2874
3. Dalam simulasi OPF dengan dua fungsi obyektif, Cuckoo memberikan nilai fitness yang paling baik dibandingkan dengan metode PSO. Nilai fitness Cuckoo pada simulasi *Economic Emission* dispatch mencapai 765,0801.
4. Cuckoo merupakan metode yang sangat baik untuk menyelesaikan permasalahan optimasi multiobjective OPF dibandingkan PSO.

5. 2. Saran

1. Penelitian ini diharapkan dapat digunakan dalam real system yang berhubungan dengan *Optimal Power Flow* (OPF).
2. Untuk penelitian multiobjective OPF perlu dikembangkan dengan penambahan *security constraint* secara lebih kompleks sehingga dapat diimplementasikan pada real sistem
3. Penelitian ini perlu dilakukan beberapa perubahan (modifikasi) untuk meningkatkan performa dari *Cuckoo Search Algorithm*

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Allen J. Wood, Bruce F. Wollenberg, Power Generation, Operation and Control, John Wiley & Sons, 1984.
- [2] Hadi Saadat, Power System Analysis, second edition. McGraw Hill, 2004.
- [3] Jizhong Zhu, PhD, Optimization of Power System Operation, 2009.
- [4] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks. IEEE Service Center, Piscataway, 1995.
- [5] Mary B. Cain, Richard P. O'Neill, Anya Castillo, History of Optimal Power Flow and Formulations, 2012.
- [6] Xin-She Yang, Engineering Optimization, An Introduction with Metaheuristic Applications, John Wiley & Sons, 2010.
- [7] J. Hazra, A. K. Sinha, A Multi-Objective Optimal Power Flow Using Particle Swarm Optimization, 2010.
- [8] M. A. Abido, "Optimal Power Flow Using Particle Swarm Optimization" *International Journal of Electrical Power and Energy Systems*, Vol. 24, No. 7, October 2002, pp. 563-571.
- [9] P. Venkatesh, K. Y. Lee, Multi-Objective Evolutionary Programming for Economic Emission Dispatch problem, *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, 2008, pp. 1-8.
- [10] Xin-She Yang, Cuckoo Search via Lévy Flights, 2010.
- [11] Xin-She Yang, *Engineering Optimisation by Cuckoo Search*, 2010.
- [12] Xin-She Yang, *Multiobjective cuckoo search for design optimization*, 2010.
- [13] Mahor A., Prasad V., Rangnekar S., *Emission Dispatch Using Particle Swarm Optimization: a Review*, Renewable and Sustainable Energy Reviews, 2009, 13, p. 2134-2141.
- [14] Jong-Bac Park, A Particle Swarm Optimization for Emission Dispatch With Nonsmooth Cost Functions, 2005

Halaman ini sengaja dikosongkan

LAMPIRAN

Validasi *Weighted Sum Method*

Metode Cuckoo

No	W1	W2	Biaya Pembangkitan	Emisi	Fitness
1	0	1	898,89	431,908	662,551
2	0,1	0,9	895,21	432,716	686,993
3	0,2	0,8	881,03	432,877	707,438
4	0,3	0,7	875,50	434,163	728,858
5	0,4	0,6	865,82	436,647	748,222
6	0,5	0,5	849,45	442,889	764,422
7	0,6	0,4	841,46	450,232	781,410
8	0,7	0,3	827,15	465,052	793,024
9	0,8	0,2	819.,4	521,654	807,775
10	0,9	0,1	808,62	521,654	807,775
11	1	0	803,47	604,865	803,468

Metode PSO

No	W1	W2	Biaya Pembangkitan	Emisi	Fitness
1	0	1	883,07	453,773	696,0942
2	0,1	0,9	901,79	433,133	688,1665
3	0,2	0,8	866,29	442,218	715,9515
4	0,3	0,7	861,56	447,636	736,7749
5	0,4	0,6	864,43	449,571	759,5610
6	0,5	0,5	854,24	446,573	769,6438
7	0,6	0,4	840,08	454,177	788,871
8	0,7	0,3	835,88	460,270	796,9330
9	0,8	0,2	818,02	497,366	807,0122
10	0,9	0,1	816,92	521,157	816,9698
11	1	0	811,76	558,738	811,7648

Listing MATLAB yang digunakan untuk *Cuckoo Algorithm* sebagai berikut :

```
clear all;
clc;
close all;

%input parameter nest
cuckoo=[20 0.5];
n      =cuckoo(1);    %population of neast
pa     =cuckoo(2);    %discovery of allien eggs

% Parameter MultiObjective
w1 = input ('weight 1= ');          % Pembobot
Cost
w2 = input ('weight 2= ');          % Pembobot
emisi

% Inisialisasi array
nd = 25;          % Pembangkit

% Input Parameter OPF
Ub = [200 80 50 35 30 40 1.05 1.1 1.1 1.1
1.1 1.1 1.1 1.1 1.1 1.1 5 5 5
5 5 5 5 5 5 ]; % Upper bounds
Lb = [50 20 15 10 10 12 0.95 0.95 0.95 0.95
0.95 0.95 0.95 0.95 0.95 0.95 2 2 2
2 2 2 2 2 2 ];% Lower bounds
par_OPF = 25;
it      = 1;
time_tic = tic;

% Inisialisasi Jumlah Iterasi
iter_max =50
0;
N_iter=0;
```

```

% Looping sederhana pembentuk nest (n,par_OPF)
for ir=1:n
    for is = 1:nd
        nest(ir,is)      = Lb(is)+rand*(Ub(is)-
Lb(is));
    end
end

%---Buat Grafik MultiObjective
hfig1 = figure(1);
hold on
title('Grafik OPF dengan Cuckoo');
set(hfig1, 'position', [10,400,600,300]);
set(hfig1, 'DoubleBuffer', 'on');
hbestplot1 = plot(1:iter_max,zeros(1,iter_max));
htext1 = text(0.6*iter_max,30,sprintf('Multi
Objective : %f', 0.0));
xlabel('Iterasi');
ylabel('MultiObjective');
hold off
drawnow;

%---Buat Grafik ED
hfig2 = figure(2);
hold on
title('Biaya Pembangkitan')
set(hfig2, 'position', [760,400,600,300]);
set(hfig2, 'DoubleBuffer', 'on');
hbestplot2 = plot(1:iter_max,zeros(1,iter_max));
htext2 = text(0.6*iter_max,30,sprintf('Biaya
Pembangkitan : %f', 0.0));
xlabel('Iterasi');
ylabel('Nilai Biaya Pembangkitan');
hold off
drawnow;

%---Buat Grafik Losses
hfig3 = figure(3);
hold on

```

```

title('emisi')
set(hfig3, 'position', [10,20,600,300]);
set(hfig3, 'DoubleBuffer', 'on');
hbestplot3 = plot(1:iter_max,zeros(1,iter_max));
htext3 = text(0.6*iter_max,30,sprintf('emisi :
%f', 0.0));
xlabel('Iterasi');
ylabel('Nilai emisi');
hold off
drawnow;

% % Inisialisasi Populasi
% for i = 1:n,
% membentuk inisialisasi nilai fitness
% Fitness = rand(n,d);
% end

% Find the current best
% n=size(nest,1);
% K=rand(size(nest))>pa;
% fitness=10^10*ones(n,1);
fitness = 10^10*zeros(n,1);
% [fmin,K]=min(fitness) ;
% best=nest(K,:);

% masuk ke load flow untuk menghitung rugi-rugi
daya
for ix=1:n;
    newnest(ix,:) = nest (ix,:) ;
    ieee30bus; % Membaca data Bus dan Jaring
Transmisi

    % Cek Constrain Max-Min
    if newnest(ix,1) >= Ub(1,1) %Slack Bus
        newnest(ix,1) = Ub(1,1);
    end
    if newnest(ix,2) >= Ub(1,2)
        newnest(ix,2) = Ub(1,2);

```

```

end
if newnest(ix,2) <= Lb(1,2)
    newnest(ix,2) = Lb(1,2);
end
if newnest(ix,3) >= Ub(1,3)
    newnest(ix,3) = Ub(1,3);
end
if newnest(ix,3) <= Lb(1,3)
    newnest(ix,3) = Lb(1,3);
end
if newnest(ix,4) >= Ub(1,4)
    newnest(ix,4) = Ub(1,4);
end
if newnest(ix,4) <= Lb(1,4)
    newnest(ix,4) = Lb(1,4);
end
if newnest(ix,5) >= Ub(1,5)
    newnest(ix,5) = Ub(1,5);
end
if newnest(ix,5) <= Lb(1,5)
    newnest(ix,5) = Lb(1,5);
end
if newnest(ix,6) >= Ub(1,6)
    newnest(ix,6) = Ub(1,6);
end
if newnest(ix,6) <= Lb(1,6)
    newnest(ix,6) = Lb(1,6);
end
if newnest(ix,7) >= Ub(1,7)
    newnest(ix,7) = Ub(1,7);
end
if newnest(ix,7) <= Lb(1,7)
    newnest(ix,7) = Lb(1,7);
end
if newnest(ix,8) >= Ub(1,8)
    newnest(ix,8) = Ub(1,8);
end
if newnest(ix,8) <= Lb(1,8)
    newnest(ix,8) = Lb(1,8);
end

```

```

if newnest(ix,9) >= Ub(1,9)
    newnest(ix,9) = Ub(1,9);
end
if newnest(ix,9) <= Lb(1,9)
    newnest(ix,9) = Lb(1,9);
end
if newnest(ix,10) >= Ub(1,10)
    newnest(ix,10) = Ub(1,10);
end
if newnest(ix,10) <= Lb(1,10)
    newnest(ix,10) = Lb(1,10);
end
if newnest(ix,11) >= Ub(1,11)
    newnest(ix,11) = Ub(1,11);
end
if newnest(ix,11) <= Lb(1,11)
    newnest(ix,11) = Lb(1,11);
end
if newnest(ix,12) >= Ub(1,12)
    newnest(ix,12) = Ub(1,12);
end
if newnest(ix,12) <= Lb(1,12)
    newnest(ix,12) = Lb(1,12);
end
if newnest(ix,13) >= Ub(1,13)
    newnest(ix,13) = Ub(1,13);
end
if newnest(ix,13) <= Lb(1,13)
    newnest(ix,13) = Lb(1,13);
end
if newnest(ix,14) >= Ub(1,14)
    newnest(ix,14) = Ub(1,14);
end
if newnest(ix,14) <= Lb(1,14)
    newnest(ix,14) = Lb(1,14);
end
if newnest(ix,15) >= Ub(1,15)
    newnest(ix,15) = Ub(1,15);
end
if newnest(ix,15) <= Lb(1,15)

```



```

        newnest(ix,15)  = Lb(1,15);
end
if newnest(ix,16) >= Ub(1,16)
    newnest(ix,16)  = Ub(1,16);
end
if newnest(ix,16) <= Lb(1,16)
    newnest(ix,16)  = Lb(1,16);
end
if newnest(ix,17) >= Ub(1,17)
    newnest(ix,17)  = Ub(1,17);
end
if newnest(ix,17) <= Lb(1,17)
    newnest(ix,17)  = Lb(1,17);
end
if newnest(ix,18) >= Ub(1,18)
    newnest(ix,18)  = Ub(1,18);
end
if newnest(ix,18) <= Lb(1,18)
    newnest(ix,18)  = Lb(1,18);
end
if newnest(ix,19) >= Ub(1,19)
    newnest(ix,19)  = Ub(1,19);
end
if newnest(ix,19) <= Lb(1,19)
    newnest(ix,19)  = Lb(1,19);
end
if newnest(ix,20) >= Ub(1,20)
    newnest(ix,20)  = Ub(1,20);
end
if newnest(ix,20) <= Lb(1,20)
    newnest(ix,20)  = Lb(1,20);
end
if newnest(ix,21) >= Ub(1,21)
    newnest(ix,21)  = Ub(1,21);
end
if newnest(ix,21) <= Lb(1,21)
    newnest(ix,21)  = Lb(1,21);
end
if newnest(ix,22) >= Ub(1,22)
    newnest(ix,22)  = Ub(1,22);

```

```

end
if newnest(ix,22) <= Lb(1,22)
    newnest(ix,22) = Lb(1,22);
end
if newnest(ix,23) >= Ub(1,23)
    newnest(ix,23) = Ub(1,23);
end
if newnest(ix,23) <= Lb(1,23)
    newnest(ix,23) = Lb(1,23);
end
if newnest(ix,24) >= Ub(1,24)
    newnest(ix,24) = Ub(1,24);
end
if newnest(ix,24) <= Lb(1,24)
    newnest(ix,24) = Lb(1,24);
end
if newnest(ix,25) >= Ub(1,25)
    newnest(ix,25) = Ub(1,25);
end
if newnest(ix,25) <= Lb(1,25)
    newnest(ix,25) = Lb(1,25);
end

% Input Cuckoo ke Bus Data Untuk Load Flow
% P Pembangkitan
busdata(1,7) = newnest(ix,1);
busdata(2,7) = newnest(ix,2);
busdata(5,7) = newnest(ix,3);
busdata(8,7) = newnest(ix,4);
busdata(11,7) = newnest(ix,5);
busdata(13,7) = newnest(ix,6);
% Tegangan Generator
busdata(1,3) = newnest(ix,7);
busdata(2,3) = newnest(ix,8);
busdata(5,3) = newnest(ix,9);
busdata(8,3) = newnest(ix,10);
busdata(11,3) = newnest(ix,11);
busdata(13,3) = newnest(ix,12);
% Tap setting Trafo

```

```

linedata(11,6) = newnest(ix,13);
linedata(12,6) = newnest(ix,14);
linedata(15,6) = newnest(ix,15);
linedata(36,6) = newnest(ix,16);
% Shunt Capacitor
busdata(10,11) = newnest(ix,17);
busdata(12,11) = newnest(ix,18);
busdata(15,11) = newnest(ix,19);
busdata(17,11) = newnest(ix,20);
busdata(20,11) = newnest(ix,21);
busdata(21,11) = newnest(ix,22);
busdata(23,11) = newnest(ix,23);
busdata(24,11) = newnest(ix,24);
busdata(29,11) = newnest(ix,25);

%% Batasan Konstrain Daya Aktif
% Batasan Maksimum Slack Bus
if (sum(busdata(:,5)) >
busdata(2,7)+busdata(5,7)+busdata(8,7)+busdata(1
1,7)+busdata(13,7)+ 200),
    fitness(ix) = 10000;
end;
%% Batasan Equality
if (sum(busdata(:,5)) ~=
busdata(1,7)+busdata(2,7)+busdata(5,7)+busdata(8
,7)+busdata(11,7)+busdata(13,7)),
    fitness(ix) = 10000;
end;
%% Batasan Inequality
if (sum(busdata(:,5)) >
busdata(1,7)+busdata(2,7)+busdata(5,7)+busdata(8
,7)+busdata(11,7)+busdata(13,7)),
    fitness(ix) = 10000;
else
    %% LOAD FLOW NEWTON RAPHSON
    [MultiObjective totalcost totalemisi] =
alirandaya (busdata,linedata)
    % Fitt(ix)      = test;
    fitness(ix) = MultiObjective;

```

```

end
fnew(ix)=fitness(ix)
if fnew<=fitness(ix),
    fitness(ix)=fnew(ix);
    nest(ix,:)=newnest(ix,:);
end
end

[fmin,K]=min(fitness) ;
best=nest(K,:);

while it <= iter_max
    % % Generate new solutions (but keep the
    current best)
    % new_nest=get_cuckoos(nest,bestnest,Lb,Ub);
    n2=size(nest,1);
    % Levy exponent and coefficient
    beta=3/2;

    sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta-1)/2)))^(1/beta);

    for j=1:n,
        s=nest(j,:);
        % This is a simple way of implementing Levy
        flights
        % For standard random walks, use step=1;
        %% Levy flights by Mantegna's algorithm
        u=randn(size(s))*sigma;
        v=randn(size(s));
        step=u./abs(v).^(1/beta);
        stepsize=0.98*step.*(s-best);
        s=s+stepsize.*randn(size(s));

        % Apply simple bounds/limits
        % nest(j,:)=simplebounds(s,Lb,Ub) % Akhirnya
        pakai fungsi.... :(
    end
end

```

```

for ix=1:n;
    newnest(ix,:) = nest (ix,:) ;
    ieee30bus; % Membaca data Bus dan Jaring
Transmisi
    % Input Cuckoo ke Bus Data Untuk Load Flow
    % P Pembangkitan
    busdata(1,7)    = newnest(ix,1) ;
    busdata(2,7)    = newnest(ix,2) ;
    busdata(5,7)    = newnest(ix,3) ;
    busdata(8,7)    = newnest(ix,4) ;
    busdata(11,7)   = newnest(ix,5) ;
    busdata(13,7)   = newnest(ix,6) ;
    % Tegangan Generator
    busdata(1,3)    = newnest(ix,7);
    busdata(2,3)    = newnest(ix,8);
    busdata(5,3)    = newnest(ix,9);
    busdata(8,3)    = newnest(ix,10);
    busdata(11,3)   = newnest(ix,11);
    busdata(13,3)   = newnest(ix,12);
    % Tap setting Trafo
    linedata(11,6)  = newnest(ix,13);
    linedata(12,6)  = newnest(ix,14);
    linedata(15,6)  = newnest(ix,15);
    linedata(36,6)  = newnest(ix,16);
    % Shunt Capacitor
    busdata(10,11)  = newnest(ix,17);
    busdata(12,11)  = newnest(ix,18);
    busdata(15,11)  = newnest(ix,19);
    busdata(17,11)  = newnest(ix,20);
    busdata(20,11)  = newnest(ix,21);
    busdata(21,11)  = newnest(ix,22);
    busdata(23,11)  = newnest(ix,23);
    busdata(24,11)  = newnest(ix,24);
    busdata(29,11)  = newnest(ix,25);

    %% Batasan Konstrain Daya Aktif
    % Batasan Maksimum Slack Bus

```

```

        if (sum(busdata(:,5)) >
busdata(2,7)+busdata(5,7)+busdata(8,7)+busdata(1
1,7)+busdata(13,7)+ 200),
            fitness(ix) = 10000;
        end;
        %% Batasan Equality
        if (sum(busdata(:,5)) ~=
busdata(1,7)+busdata(2,7)+busdata(5,7)+busdata(8
,7)+busdata(11,7)+busdata(13,7)),
            fitness(ix) = 10000;
        end;
        %% Batasan Inequality
        if (sum(busdata(:,5)) >
busdata(1,7)+busdata(2,7)+busdata(5,7)+busdata(8
,7)+busdata(11,7)+busdata(13,7)),
            fitness(ix) = 10000;
        else
            %% LOAD FLOW NEWTON RAPHSON
            [MultiObjective totalcost totalemisi] =
alirandaya (busdata,linedata)
            % Fitt(ix) = test;
            fitness(ix) = MultiObjective;
        end
        fnew(ix)=fitness(ix) ;
        if fnew<=fitness(ix),
            fitness(ix)=fnew(ix);
            nest(ix,:)=newnest(ix,:);
        end
    end
end

[fmin,K]=min(fitness) ;
best=nest(K,:);

n=size(nest,1) ;
% Discovered or not -- a status vector
K=rand(size(nest))>pa;

% New solution by biased/selective random walks

```

```

% stepsize=rand*(nest(randperm(n),:)-
nest(randperm(n),:)) % Original
stepsize=rand*(nest(sort(n))-
nest(randperm(n),:)) ;

newnest=newnest+stepsize.*K ;% newnest or
nest.....

% % Grafik Evaluasi Nilai Fitness
plotvector1 = get(hbestplot1,'YData');
plotvector1(it) = fmin ;
set(hbestplot1,'YData',plotvector1);
set(htext1,'String',sprintf('Fungsi Objektif:
%f', fmin));
drawnow

% % Grafik Cost Dispatch
plotvector2=get(hbestplot2,'Ydata');
plotvector2(it) = totalcost(1) ;
set(hbestplot2,'Ydata',plotvector2);
drawnow

% % Grafik Emisi
plotvector4 = get(hbestplot3,'YData');
plotvector4(it) = totalemisi(1);
set(hbestplot3,'YData',plotvector4);
drawnow

it=it+1 ;
end %% End of iterations
HasilAkhir2

```

RIWAYAT PENULIS



Herbanindya Febri Trusatmaji dilahirkan di Semarang, 12 Februari 1993. Ayah saya bernama Wilaraso Hermanto bekerja sebagai konsultan bangunan dan ibu saya bernama Sri Wibawanti bekerja sebagai karyawan Rumah Sakit di Semarang. Penulis menempuh jenjang pendidikan dari SD. H. Isriati Semarang, dan melanjutkan ke SMPN 3 Semarang. Kemudian penulis melanjutkan ke SMAN 4 Semarang hingga lulus tahun 2010. Pada tahun yang sama penulis melanjutkan studinya ke Institut Teknologi Sepuluh Nopember Surabaya Fakultas Teknologi Industri Jurusan Teknik Elektro dan mengambil bidang studi Teknik Sistem Tenaga.

Penulis dapat dihubungi melalui email :
Trusatmaji@yahoo.com

Halaman ini sengaja dikosongkan